



## VPC-SI System StarOS リリース 21.28 アドミニストレーション ガイド

初版：2022年9月29日

最終更新：2023年3月2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/>

【注意】 シスコ製品をご使用になる前に、安全上の注意（ [www.cisco.com/jp/go/safety\\_warning/](http://www.cisco.com/jp/go/safety_warning/) ）をご確認ください。本書は、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。また、契約等の記述については、弊社販売パートナー、または、弊社担当者にご確認ください。

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at [www.cisco.com/go/offices](http://www.cisco.com/go/offices).

The documentation set for this product strives to use bias-free language. For purposes of this documentation set, bias-free is defined as language that does not imply discrimination based on age, disability, gender, racial identity, ethnic identity, sexual orientation, socioeconomic status, and intersectionality. Exceptions may be present in the documentation due to language that is hardcoded in the user interfaces of the product software, language used based on standards documentation, or language that is used by a referenced third-party product.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: <https://www.cisco.com/c/en/us/about/legal/trademarks.html>. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

© 2022 Cisco Systems, Inc. All rights reserved.



## 目次

---

はじめに :

[このマニュアルについて](#) xxix

[使用する表記法](#) xxx

[関連資料](#) xxxii

[カスタマー サポートへの問い合わせ](#) xxxii

---

第 1 章

**VPC-SI 入門 1**

[製品の説明](#) 1

[仮想化されたモビリティ機能](#) 2

[VM インターコネクト アーキテクチャ](#) 2

[スタンドアロンインスタンス](#) 2

[フィーチャ セット](#) 3

[インターフェイスおよびアドレッシング](#) 3

[暗号化](#) 4

[セキュリティ](#) 4

[冗長性と可用性](#) 5

[プラットフォーム要件](#) 5

[ICSR のサポート](#) 5

[IFTASK 冗長性](#) 6

[ハイパーバイザ要件](#) 6

[VM の設定](#) 7

[vCPU と vRAM のオプション](#) 7

[vNIC のオプション](#) 8

[ハードドライブストレージ](#) 8

[DPDK 内部フォワーダ](#) 8

キャパシティ、CEPS、およびスループット	11
診断およびモニタリング	12
StarOS VPC-SI のビルドコンポーネント	12
VPC-SI ブートパラメータ	12
起動パラメータファイルのフォーマット	13
ネットワーク インターフェイス ID	13
VPC-SI ブートパラメータの設定	18
VNFM インターフェイスオプションの設定	21
VPP 設定パラメータ	23
ソフトウェアインストールおよびネットワーク展開	27

---

**第 2 章****システムの運用と設定 29**

用語	29
コンテキスト	29
論理インターフェイス	30
管理インターフェイス	30
バインディング	30
サービス	31
AAA サーバー	31
サブスクライバ	32
システムがコンテキストを選択する方法	33
コンテキストレベルの管理ユーザーセッションのコンテキスト選択	33
サブスクライバセッションのコンテキスト選択	36
コンフィギュレーションファイルの概要	36
IP アドレスの表記法	37
IPv4 ドット付き 10 進表記	38
IPv6 コロン区切り 16 進表記	38
CIDR 表記	38
英数字の文字列	39
文字セット	39
引用符付き文字列	41

## 第 3 章

## 使用する前に 43

- StarOS CLI を使用した初期設定 43
- システム管理ユーザーの設定 45
  - 同時 CLI セッション数の制限 45
  - CLI セッションの自動ログアウト 46
- リモートアクセス用のシステムの設定 47
- SSH オプションの設定 49
  - SSH ホストキー 49
    - SSH キーのサイズ設定 49
    - SSH キー生成の待機時間の設定 50
    - SSH 暗号化暗号方式の指定 50
    - MAC アルゴリズムの設定 51
    - SSH キーの生成 53
    - SSH キーペアの設定 54
  - 承認済み SSH ユーザーアクセス 54
    - SSH ユーザーアクセスの認可 54
  - SSH ユーザーログインの制限事項 55
    - 許可済みユーザーリストの作成 55
  - SSH ユーザーログイン認証 56
  - セキュアなセッションログアウト 57
    - デフォルトの sshd セキュア セッション ログアウト パラメータの変更 58
  - 外部サーバーへの SSH クライアントログイン 58
    - SSH クライアント暗号の設定 58
    - 優先認証方式の設定 59
    - SSH クライアントキーペアの生成 60
    - 外部サーバーへの SSH クライアント公開キーのプッシュ 60
  - NETCONF の有効化 61
- 2 番目の IP アドレスを使用した管理インターフェイスの設定 61
- Open SSH から Cisco SSH へのアップグレードと移行 62
  - 機能の概要と変更履歴 62

変更された機能 63

VM ハードウェアの検証 65

## 第 4 章

### システム設定 67

インターフェイスとポートの設定の確認と保存 67

システムタイミングの設定 68

システムクロックとタイムゾーンの設定 68

クロックとタイムゾーンの設定の確認と保存 69

ネットワーク時間プロトコルのサポートの設定 69

ローカルソースを使用した NTP サーバーの設定 70

タグ付きインターフェイスでの NTP の設定 71

ロードバランサの使用 71

NTP の設定確認 71

ソフトウェア RSS の設定 73

DI ネットワーク RSS の暗号化 73

機能の概要と変更履歴 73

変更された機能 74

コマンドの変更 74

SF ブート設定ポーズの設定 74

CLI タイムスタンプの有効化 75

CLI 確認プロンプトの設定 75

自動確認の有効化 76

autoconfirm コマンドと configure コマンドに対する確認の要求 76

特定の Exec モードのコマンドに対する確認の要求 77

システム管理ユーザーの設定 78

ユーザー名の文字制限 78

コンテキストレベルの管理ユーザーの設定 79

コンテキストレベルのセキュリティ管理者の設定 80

コンテキストレベルの管理者の設定 80

コンテキストレベルのオペレータの設定 80

コンテキストレベルのインスペクタの設定 81

システムと LI 設定の分離	81
コンテキストレベルの管理ユーザー設定の確認	83
ローカルユーザーの管理ユーザーの設定	83
ローカルユーザー設定の確認	84
ローカルユーザーデータベースの更新	84
ローカルユーザーデータベースの更新とダウングレード	84
指定されたルートディレクトリへのユーザーアクセスの制限	85
SFTP ルートディレクトリの設定	85
SFTP ルートディレクトリとローカルユーザーの関連付け	86
SFTP ルートディレクトリと管理者の関連付け	86
SFTP ルートディレクトリと設定管理者の関連付け	86
システム管理ユーザー用の TACACS+ の設定	86
動作	86
ユーザーアカウントの要件	88
TACACS+ ユーザーアカウントの要件	88
StarOS ユーザーアカウントの要件	88
TACACS+ AAA サービスの設定	89
非ローカル VPN 認証のための TACACS+ の設定	90
TACACS+ の設定の確認	90
TACACS+ サーバーでの IPv6 アドレスサポート	91
認証方式の分離	91
コンソールの TACACS+ 認証の無効化	91
コンソールの AAA ベースの認証を無効化	92
コンテキストレベルで TACACS+ 認証を無効化	92
コンソール/VTY 回線でのローカルユーザーログインの制限	93
AAA ベースのユーザーのコンソールアクセスの制限	93
設定変更の確認	94
シャーシキーの設定	94
概要	94
新しいシャーシキーの値の設定	95
CLI コマンド	95

クイック セットアップ ウィザード	96
FSC ファブリックの自動リセットの有効化	96

---

第 5 章	<b>モードロックメカニズムの設定</b>	99
	モードロック設定の概要	99
	Exclusive-Lock の要求	100
	URL スクリプトに対するロック設定の効果	102
	設定ファイルの保存	102
	リロードコマンドとシャットダウンコマンド	103
	show administrators コマンド	104

---

第 6 章	<b>管理設定</b>	105
	SNMP MIB ブラウザ	105
	SNMP サポート	107
	SNMP とアラームサーバーパラメータの設定	108
	SNMP パラメータの確認	109
	SNMP トラップ生成の制御	110

---

第 7 章	<b>設定の確認と保存</b>	111
	設定の確認	111
	機能設定	111
	サービス構成	112
	コンテキストの設定	112
	システム設定 (System Configuration)	112
	設定エラーの検出	113
	ファイルシステムの同期	113
	設定の保存	113

---

第 8 章	<b>システムインターフェイスとポート</b>	115
	コンテキスト	115
	コンテキストの作成	115



コンテキストの表示と確認	116
イーサネットインターフェイスとポート	116
インターフェイスの作成	117
ポートの設定とインターフェイスへのバインド	117
インターフェイスのスタティックルートの設定	118
ポート設定の表示と確認	118
VLANs	119
ハイパーバイザ	119
VLAN および管理ポート	120

---

**第 9 章**

<b>システムセキュリティ</b>	<b>121</b>
パスワードの保護	121
セキュアなパスワードの暗号化	121
最新ではない暗号化および復号化のサポート	122
ICSR 設定のサポート	123
暗号化された SNMP コミュニティストリング	123
強化されたパスワードセキュリティ	123
合法的傍受の制約事項	123
LI サーバーアドレス	123
代行受信の変更	124
ユーザーの追加、変更、削除	124
ユーザーの追加または削除の通知	125
権限レベル変更の通知	125
オペレーティング システム シェルへのユーザーアクセス	125
Test-Commands	125
cli test-commands モードの有効化	126
CLI テストコマンドにアクセスするためのパスワードの有効化	126
Exec モードの cli test-commands	127
コンフィギュレーション モードの cli test-commands	127
暗号化のための COTS ハードウェアの使用	128
OS とプラットフォームの乱数生成器のサポート	129

機能の概要と変更履歴 129

機能説明 130

---

第 10 章

セキュアなシステム コンフィギュレーション ファイル 133

機能の概要と変更履歴 133

機能説明 134

システム コンフィギュレーション ファイルの保護方法 134

デジタル署名の作成 134

デジタル署名の検証 135

署名検証の設定 135

検証用の RSA 公開キーのインポート 135

署名検証の有効化または無効化 136

---

第 11 章

ソフトウェア管理の運用 137

ローカルファイルシステムの概要 137

ローカルファイルシステムで使用されるファイルタイプ 137

boot.sys ファイルの概要 138

ローカルファイルシステムの保守 138

ファイルシステム管理コマンド 139

ディレクトリの作成 139

ファイルとディレクトリの名前の変更 139

ファイルのコピー 139

ファイルの削除 140

ディレクトリの削除 140

ローカルデバイスのフォーマット 140

既存の CLI コンフィギュレーション ファイルの適用 141

ローカルファイルシステム上のファイルの表示 142

ローカルデバイスの内容の表示 142

CLI 設定と boot.sys ファイルの表示 142

オペレーティングシステムのファイルの検証 142

Elastic Services Controller のクラウド初期化サポート 143

起動スタックの設定	143
システムの起動方式	144
現在の起動スタックの表示	144
新しい起動スタックエントリの追加	145
起動スタックエントリの削除	146
オペレーティング システム ソフトウェアのアップグレード	146
前提条件	147
AutoVNF、CF、ESC、および UEM の VIP アドレスの取得	147
OS リリースバージョンとビルド番号の識別	149
サポートサイトからソフトウェアイメージをダウンロード	150
Zookeeper データベースの確認	151
ESC データベースの確認	152
/flash デバイスの空き領域の確認	152
StarOS イメージを /flash に転送	153
現在の設定ファイルのコピーの保存	155
リリース 20.0 からのダウングレード	155
オフラインソフトウェアのアップグレード	156
新規コールポリシーの設定	156
Day バナーのメッセージの設定	157
現在の CLI コンフィギュレーション ファイルのバックアップ	157
実行コンフィギュレーションの保存	158
新しい起動スタックエントリの作成	159
システムの再起動	160
実行コンフィギュレーションの保存	164
ファイルシステムの同期	166
システムの再起動	167
以前のソフトウェアイメージの復元	172
ライセンスキーの管理	172
新しいシステムライセンスキー	172
セッション使用とライセンス使用のライセンス	173
新しいライセンスキーのインストール	173

キーのカットアンドペースト	173
ライセンスキーをコンフィギュレーションファイルに追加	174
ライセンスの期限切れの動作	175
ライセンスキーの要求	175
ライセンス情報の表示	175
ライセンスキーの削除	175
ローカルユーザー管理アカウントの管理	175
ローカルユーザーパスワードのプロパティの設定	176
ローカルユーザーのアカウント管理プロパティの設定	176
ローカルユーザーアカウントのロックアウト	176
ローカルユーザーアカウントの一時停止	177
ローカルユーザーパスワードの変更	177

## 第 12 章

## スマートライセンス 179

機能の概要と変更履歴	179
スマート ソフトウェア ライセンシング	180
Cisco Smart Software Manager	181
スマートアカウントおよびバーチャルアカウント	182
Cisco スマートアカウントの要求	182
ソフトウェアタグと権限付与タグ	183
スマート ライセンスの設定	185
スマートライセンシングのモニタリングとトラブルシューティング	186
スマートライセンシングのバルク統計情報	187

## 第 13 章

## システムのモニタリング 189

SNMP 通知	190
システムのステータスとパフォーマンスのモニタリング	190
統計情報とカウンタのクリア	192

## 第 14 章

## モニタープロセスのリスト 193

機能の概要と変更履歴	193
------------	-----

機能説明	194
モニターリングおよびトラブルシューティング	194
コマンドや出力の表示	194
show process status	194

---

**第 15 章**

<b>バルク統計情報</b>	<b>197</b>
機能の概要と変更履歴	197
収集サーバーとの通信の設定	198
標準の設定	198
オプションの設定	199
バルク統計情報スキーマの設定	199
個別のバルク統計情報設定ファイルの設定	200
show bulkstats コマンドの使用	201
設定の確認	201
コンフィギュレーションの保存	202
収集されたバルク統計情報データの表示	202
SSD でのバルク統計情報サンプルの収集	202
SFTP 公開キー認証	203
機能説明	203
SFTP 公開キー認証	204
バルク統計情報の手動収集と転送	204
バルク統計のカウンタと情報のクリア	204
バルク統計情報スキーマの名称	204
統計タイプ	204
データ型	205
主な変数	206
バルク統計情報のイベントログメッセージ	208

---

**第 16 章**

<b>システム ログ</b>	<b>209</b>
機能の概要と変更履歴	209
システムログのタイプ	211

イベントロギングパラメータの設定	212
イベントログフィルタの設定	212
Exec モードのフィルタリング	212
グローバル コンフィギュレーション モードのフィルタリング	215
syslog サーバの設定	216
アクティブログの設定	217
ファシリティの指定	218
トレースロギングの設定	228
モニターログの設定	229
モニターログの有効化	229
モニターログの無効化	229
ロギング設定と統計情報の表示	230
CLI を使用したイベントログの表示	230
クラッシュログの設定と表示	231
クラッシュロギングのアーキテクチャ	232
ソフトウェアクラッシュログ接続先の設定	233
CLI を使用して要約されたクラッシュログ情報の表示	234
過剰なイベントロギングの削減	235
ログソースのしきい値の設定	235
ログのチェックポイントニング	236
ログファイルの保存	237
イベント ID の概要	237
イベントのシビラティ (重大度)	246
ログ出力のイベント ID 情報の概要	247

## 第 17 章

トラブルシューティング	249
ネットワーク接続の確認	249
ping コマンドまたは ping6 コマンドの使用	249
構文	250
トラブルシューティング	250
traceroute または traceroute6 コマンドの使用	251

tracert	251
tracert6	251
IP ルートの表示	251
アドレス解決プロトコルテーブルの表示	252
システム診断ユーティリティの使用	252
モニターユーティリティの使用	253
プロトコルモニターの使用	253
特定サブスクリバのプロトコルモニターの使用	254
SSD の生成	256
サポートデータコレクターの設定と使用	257
ハイパーバイザによる強制再起動の開始	257

---

## 第 18 章

パケットキャプチャ (PCAP) トレース	259
機能情報	259
機能説明	260
PCAP トレースの設定	261
CDRMOD の複数のインスタンスの有効化	261
16 進ダンプモジュールの設定	261
16 進ダンプファイルパラメータの設定	264
16 進ダンプの有効化または無効化	267
MME の PCAP トレースの有効化	268
PCAP トレースのモニタリングとトラブルシューティング	269
コマンドや出力の表示	269
show cdr statistics	269
show { hexdump-module   cdr } file-space-usage	270
show hexdump-module statistics	271

---

## 第 19 章

システム リカバリ	275
前提条件	275
コンソールアクセス	275
起動イメージ	275

ブート CLI へのアクセス	276
再起動の開始	276
Enter CLI Mode	277
boot コマンドシンタックス	277
起動できないシステムからの復旧	277

## 第 20 章

## アクセス コントロール リスト 279

概要	279
ACL の概要	280
ルール	280
アクション	280
基準	281
ルールの順序	282
システム上での ACL の設定	282
ACL の作成	283
サブスクライバトラフィックのアクションと基準の設定	283
未定義の ACL の設定	284
ACL 設定の確認	284
IP ACL の適用	285
インターフェイスへの ACL の適用	286
個々のインターフェイスへの ACL の適用	287
インターフェイス上の ACL 設定の確認	287
コンテキストへの ACL の適用	288
コンテキスト内のすべてのトラフィックへの ACL の適用	289
コンテキストでの ACL 設定の確認	289
RADIUS ベースのサブスクライバにおける ACL の適用	290
個々のサブスクライバへの ACL の適用	290
個々のサブスクライバへの ACL 設定の確認	291
default というサブスクライバへの ACL の適用	292
default というサブスクライバへの ACL の適用	292
default というサブスクライバに対する ACL 設定の確認	292



サービス指定のデフォルトのサブスライバへの ACL の適用	293
サービス指定のデフォルトのサブスライバへの ACL の適用	294
サービス指定のデフォルトのサブスライバへの ACL 設定の確認	294
複数のサブスライバへの単一 ACL の適用	295
複数のサブスライバへの APN を介した ACL の適用	296

---

## 第 21 章

### 輻輳制御 299

概要 299

輻輳制御の設定 300

輻輳制御のしきい値の設定 300

サービス輻輳ポリシーの設定 301

MME でのオーバーロードレポートの設定 302

輻輳制御リダイレクト オーバーロード ポリシーの有効化 302

サービス オーバーロード ポリシーの検証 303

輻輳制御の設定の確認 303

MME の輻輳アクションプロファイルの確認 303

コール時間または非アクティブ時間に基づくサブスライバの接続解除 303

---

## 第 22 章

### ルーティング 305

ルーティング ポリシー 305

IP プレフィックスリストの作成 306

ルートアクセスリストの作成 306

AS パスアクセスリストの作成 306

ルートマップの作成 307

設定例 307

スタティック ルーティング 307

コンテキストへのスタティックルートの追加 308

コンテキストからのスタティックルートの削除 308

OSPF ルーティング 309

OSPF バージョン 2 の概要 309

OSPFv2 設定の基本 310

特定のコンテキストの OSPF ルーティングの有効化	310
特定のインターフェイスでの OSPF の有効化	311
OSPF へのルートの再配布 (オプション)	311
OSPF 設定パラメータの確認	311
OSPFv3 ルーティング	311
OSPFv3 の概要	312
OSPFv3 設定の基本	312
特定のコンテキストにおける OSPFv3 ルーティングの有効化	312
特定のインターフェイスでの OSPFv6 の有効化	312
OSPFv3 へのルートの再配布 (オプション)	313
OSPFv3 設定パラメータの確認	313
等コストマルチパス (ECMP)	313
BGP-4 ルーティング	314
BGP サポートの概要	314
BGP の設定	315
BGP へのルートの再配布 (オプション)	315
BGP コミュニティと BGP 拡張コミュニティ	316
BGP コミュニティ	316
BGP 拡張コミュニティ	317
BGP ローカルプリファレンス	318
ICSR および SRP グループ	318
スタンバイ ICSR シャーシからの BGP ルートのアドバタイジング	319
ICSR の設定可能な BGP ルートのアドバタイズメント間隔	319
BGP CLI 設定コマンド	320
BGP 設定パラメータの確認	321
BGP ピアの制限	322
機能の概要と変更履歴	322
機能説明	322
機能の仕組み	323
BGP ピア制限の設定	323
モニターリングおよびトラブルシューティング	324

双方向フォワーディング検出	325
BFD サポートの概要	325
BFD の設定	326
BFD コンテキストの設定	326
スタティックルートの IPv4 BFD の設定	326
スタティックルートの IPv6 BFD の設定	327
シングルホップ用の BFD の設定	327
マルチホップ BFD の設定	328
BFD の拡張性	328
BGP ネイバーとコンテキストの関連付け	328
OSPF ネイバーとコンテキストの関連付け	329
BFD ネイバークラスタと BFD プロトコルの関連付け	329
OSPF インターフェイスでの BFD の有効化	329
ICSR の BFD 接続のモニタリング	329
設定の保存	330
ICSR のシャーシ間 BFD のモニタリング	330
プライマリシャーシ BFD モニタリングの有効化	330
ICSR Dead 間隔を無視する BFD の設定	330
ICSR スイッチオーバー ガード タイマー の設定	331
BFD マルチホップ フォールオーバーの有効化	331
BFD 間隔の調整	333
スタンバイ ICSR シャーシからの BGP ルートのアドバタイジングの有効化	333
設定の保存	333
リンク集約メンバーリンクの BFD サポート	333
概要	333
BFD Linkagg メンバーリンクのサポートの設定	334
設定の保存	335
ルーティング情報の表示	335

オーバーラッピング IP アドレスプールのサポート : GGSN	338
RADIUS VLAN サポート : Enhanced Charging サービス	338
APN サポート : PDN ゲートウェイ (P-GW)	339
VLAN および StarOS	339
VLAN およびハイパーバイザ	339
VLAN および KVM ハイパーバイザ	340
ネットワークの分離	340
VLAN 対ブリッジインターフェイス	340
その他の情報	340
VLAN 対応 VM	341
VLAN および VMware	341
VLAN の設定	341
その他の情報	342
VLAN タグの設定	342
ポート設定の確認	343
サブスクリバ VLAN の関連付けの設定	344
使用される RADIUS 属性	344
ローカルサブスクリバのプロファイルの設定	344
サブスクリバのプロファイル設定の確認	345
VLAN 関連の CLI コマンド	345

---

第 24 章	<b>BGP MPLS VPN</b>	349
	はじめに	349
	PE に接続された MPLS-CE	350
	VPC-SI PE として	350
	概要	350
	設定例	351
	BGP MPLS VPN の IPv6 サポート	352
	概要	352
	設定例	353
	VPN 関連の CLI コマンド	356

## 第 25 章

## コンテンツサービスのステアリング 363

## 概要 363

## 内部コンテンツサービスのステアリングの設定 364

## 内部 CSS の IP アクセスリストの定義 364

## 個々のサブスクリバへの ACL の適用 (オプション) 365

## 複数のサブスクリバへの ACL の適用 (オプション) 365

## サブスクリバの名前付きデフォルトへの ACL の適用 (オプション) 365

## サービス指定のデフォルトのサブスクリバへの ACL の適用 (オプション) 365

## APN を介した複数のサブスクリバへの ACL の適用 (オプション) 366

## 第 26 章

## セッションリカバリ 367

## セッションリカバリの仕組み 367

## セッションリカバリをサポートするためのシステムの設定 370

## セッションリカバリの有効化 370

## アウトオブサービスシステムでのセッションリカバリの有効化 370

## インサービスシステムでのセッションリカバリの有効化 371

## セッションリカバリ機能の無効化 372

## セッションリカバリ ステータスの表示 372

## セッション情報回復の確認 373

## 制御タスクの統計情報のリカバリ 374

## show rct stats コマンド 374

## show rct stats verbose の出力例 375

## 第 27 章

## シャーシ間セッションリカバリ 377

## 概要 377

## シャーシ間の通信 379

## チェックポイントメッセージ 379

## SRP CLI コマンド 379

## Exec モード CLI コマンド 379

## show コマンド 380

AAA モニター	381
BGP の連携動作	381
要件	382
ICSR の運用	383
シャーシの初期化	386
シャーシの動作	386
シャーシ通信	386
シャーシスイッチオーバー	386
ICSR の設定	386
SRP チェックポイントの設定	387
SRP チェックポイントの設定	387
モニターリングおよびトラブルシューティング	388
サービス冗長性プロトコル (SRP) コンテキストの設定	388
SRP コンテキストの作成とバインド	389
SRP コンテキストパラメータの設定	389
スイッチオーバー移行の最適化	392
SRP コンテキストのインターフェイスパラメータの設定	396
SRP チェックポイントメッセージングエラーに対する NACK 生成の設定	396
LZ4 圧縮アルゴリズムの設定	398
スタンバイの ICSR シャーシによる同期時間の短縮	398
SRP 設定の確認	399
ICSR の送信元コンテキストの変更	399
BGP ルータとゲートウェイアドレスの設定	400
BGP 用の SRP コンテキストの設定	400
BGP 設定の確認	401
ICSR の接続先コンテキストの変更	401
接続先コンテキストでの BGP ルータとゲートウェイアドレスの設定	401
接続先コンテキスト用の BGP の SRP コンテキストの設定	401
サブスクリイバをデフォルトモードに設定	402
接続先コンテキストでの BGP 設定の確認	402
スタンバイシステムでのバルク統計情報収集の無効化	402

プライマリ設定とバックアップ設定の確認	402
サブスライバ状態管理の監査プロセスの設定	403
ICSR 運用のトラブルシューティング	403
オペレーティングシステムの更新	404
両方の ICSR システム	409
StarOS イメージのダウンロードと転送	409
スタンバイ ICSR システム	410
正常性チェックの実行	410
SRP チェックの実施	411
BGP チェックの実施	411
ブートレコードの更新	411
ファイルシステムの同期	412
StarOS の再起動	412
設定ファイルの更新	412
ソフトウェアバージョンの確認	412
設定ファイルの保存	412
更新プロセスの完了	413
セッションの同期の待機	413
プライマリシステム (Primary System)	413
SRP スイッチオーバーの開始	413
新しくアクティブになったシステムでの AAA モニターステータスの確認	414
ソフトウェア更新の実行	414
SRP スイッチオーバーの開始	414
テストコールの実施	415
フォールバック手順	415
第 28 章	データコレクターのサポート 417
	概要 417
	SDR 収集の設定 418
	SDR 収集設定の表示 418
	SDR 情報の収集と保存 419

レコード収集の管理	419
SDR を使用した問題の診断	421
SDR CLI コマンド	421
設定コマンド (グローバル コンフィギュレーション モード)	422
support record	422
support collection	423
Exec モードのコマンド	423
show support record	423
delete support record	424
show support collection	424

## 付録 A :

<b>エンジニアリングルール</b>	425
CLI セッションルール	425
VPC インターフェイスとポートのルール	426
vNIC イーサネットポート	426
パケットデータネットワーク (PDN) インターフェイスのルール	426
コンテキストルール	427
サブスクライバルール	430
サービスルール	431
Access Control List (ACL; アクセスコントロールリスト) のエンジニアリングルール	431
ECMP グループ	432
VPN スケーリングの要件	433

## 付録 B :

<b>StarOS のタスク</b>	439
概要	439
プライマリ タスク サブシステム	440
コントローラおよびマネージャ	442
サブシステムのタスク	442
システム開始サブシステム	443
高可用性サブシステム	444
Resource Manager サブシステム	445
仮想プライベート ネットワーク サブシステム	445



ネットワーク処理ユニットサブシステム 448

セッションサブシステム 449

プラットフォームプロセス 463

管理プロセス 467

---

付録 C :

**NETCONF および ConfD 469**

機能の概要と変更履歴 469

概要 471

ConfD の設定 472

SSH キーの要件 472

NETCONF プロトコル コンフィギュレーション モード 473

bulkstats 474

confd-user 474

kpi 474

netconf notifications events 475

netconf notifications snmp 475

netconf port 475

rest auth-policy 476

rest certificate 477

rest hostname 477

rest port 478

設定例 478

設定の確認 478

show confdmgr コマンド 479

clear confdmgr confd cdb 484

clear confdmgr statistics 485

YANG モデル 485

Show Support Details (SSD) 486

ConfD の例 486

サーバー ConfD 486

バルク統計情報 488

Exec CLI モデル 490

ConfD のアップグレードサポート	492
ECS コマンド用の CLI ベースの YANG モデル	492
CDB のシーディングと同期	492
show configuration confd コマンド	493
CDB のメンテナンス	493
clear confdmgr confd cdb	494
configure confd <url>	494
save configuration <url> confd	494
サポートされている StarOS ECS 設定コマンド	495

## 付録 D :

<b>ICSR のチェックポイント</b>	<b>497</b>
チェックポイントの概要	497
マクロチェックポイント	498
GGSN_APN ID MAPPING	498
インスタンスレベルのチェックポイント	498
SERVICE_ID MAPPING	499
VPNMGR_ID MAPPING	499
マイクロチェックポイント	499
未分類	500
SESS_UCHKPT_CMD_INVALIDATE_CRR	500
SESS_UCHKPT_CMD_UPDATE_CLPSTATS	500
SESS_UCHKPT_CMD_UPDATE_IDLESECS	500
DCCA カテゴリ	501
SESS_UCHKPT_CMD_DCCA_SESS_INFO	501
ECS カテゴリ	501
SESS_UCHKPT_CMD_ACS_CALL_INFO	501
SESS_UCHKPT_CMD_ACS_GX_LI_INFO	501
SESS_UCHKPT_CMD_ACS_SESS_INFO	502
SESS_UCHKPT_CMD_DEL_ACS_CALL_INFO	502
SESS_UCHKPT_CMD_DEL_ACS_SESS_INFO	502
SESS_UCHKPT_CMD_DYNAMIC_CHRG_CA_INFO	503
SESS_UCHKPT_CMD_DYNAMIC_CHRG_DEL_CA_INFO	503
SESS_UCHKPT_CMD_DYNAMIC_CHRG_DEL_QG_INFO	503

SESS_UCHKPT_CMD_DYNAMIC_CHRG_QG_INFO	504
SESS_UCHKPT_CMD_DYNAMIC_RULE_DEL_INFO	504
SESS_UCHKPT_CMD_DYNAMIC_RULE_INFO	504
ePDG カテゴリ	505
SESS_UCHKPT_CMD_DELETE_EPDG_BEARER	505
SESS_UCHKPT_CMD_UPDATE_EPDG_BEARER	505
SESS_UCHKPT_CMD_UPDATE_EPDG_PEER_ADDR	505
SESS_UCHKPT_CMD_UPDATE_EPDG_REKEY	506
SESS_UCHKPT_CMD_UPDATE_EPDG_STATS	506
ファイアウォール/ECS カテゴリ	506
SESS_UCHKPT_CMD_SFW_DEL_RULE_INFO	506
SESS_UCHKPT_CMD_SFW_RULE_INFO	507
GGSN カテゴリ	507
SESS_UCHKPT_CMD_GGSN_DELETE_SUB_SESS	507
SESS_UCHKPT_CMD_GGSN_UPDATE_RPR	507
SESS_UCHKPT_CMD_GGSN_UPDATE_SESSION	508
SESS_UCHKPT_CMD_GGSN_UPDATE_STATS	508
SESS_UCHKPT_CMD_UPDATE_COA_PARAMS	508
Gx インターフェイスカテゴリ	509
SESS_UCHKPT_CMD_ACS_VOLUME_USAGE	509
SESS_UCHKPT_CMD_UPDATE_SGX_INFO	509
NAT カテゴリ	509
SESS_UCHKPT_CMD_GR_UPDATE_NAT_REALM_PORT_INFO1	509
SESS_UCHKPT_CMD_GR_UPDATE_NAT_REALMS	510
SESS_UCHKPT_CMD_NAT_SIP_ALG_CALL_INFO	510
SESS_UCHKPT_CMD_NAT_SIP_ALG_CONTACT_PH_INFO	510
SESS_UCHKPT_CMD_UPDATE_DSK_FLOW_CHKPT_INFO	511
SESS_UCHKPT_CMD_UPDATE_NAT_BYPASS_FLOW_INFO	511
P-GW カテゴリ	512
SESS_UCHKPT_CMD_PGW_DELETE_SUB_SESS	512
SESS_UCHKPT_CMD_PGW_OVRCHRG_PRTCTN_INFO	512
SESS_UCHKPT_CMD_PGW_SGWRESTORATION_INFO	512
SESS_UCHKPT_CMD_PGW_UBR_MBR_INFO	512

SESS_UCHKPT_CMD_PGW_UPDATE_APN_AMBR	513
SESS_UCHKPT_CMD_PGW_UPDATE_INFO	513
SESS_UCHKPT_CMD_PGW_UPDATE_LI_PARAM	513
SESS_UCHKPT_CMD_PGW_UPDATE_PDN_COMMON_PARAM	513
SESS_UCHKPT_CMD_PGW_UPDATE_QOS	513
SESS_UCHKPT_CMD_PGW_UPDATE_SGW_CHANGE	513
SESS_UCHKPT_CMD_PGW_UPDATE_STATS	513
Rf インターフェイスカテゴリ	514
SESS_UCHKPT_CMD_ACS_ACCOUNTING_TYPE_QCI_RF	514
SESS_UCHKPT_CMD_ACS_ACCOUNTING_TYPE_QCI_RF_WITH_FC	514
SESS_UCHKPT_CMD_ACS_ACCOUNTING_TYPE_RATING_GROUP_RF	514
SESS_UCHKPT_CMD_ACS_ACCOUNTING_TYPE_RATING_GROUP_RF_WITH_FC	515
S6b インターフェイスカテゴリ	515
SESS_UCHKPT_CMD_S6B_INFO	515
SaMOG カテゴリ	515
SESS_UCHKPT_CMD_CGW_DELETE_BEARER	515
SESS_UCHKPT_CMD_CGW_DELETE_PDN	516
SESS_UCHKPT_CMD_CGW_UPDATE_BEARER_QOS	516
SESS_UCHKPT_CMD_CGW_UPDATE_PDN	516
SESS_UCHKPT_CMD_CGW_UPDATE_STATS	516
SESS_UCHKPT_CMD_CGW_UPDATE_UE_PARAM	517
SESS_UCHKPT_CMD_SAMOG_ACCT_INTERIM_INFO	517
SESS_UCHKPT_CMD_SAMOG_ACCT_START_INFO	517
SESS_UCHKPT_CMD_SAMOG_EOGRE_TUNNEL_INFO	517
SESS_UCHKPT_CMD_SAMOG_GTPV1_UPDATE_PDN_INFO	518
SESS_UCHKPT_CMD_SAMOG_HANDOFF_AUTHEN_INFO	518
SESS_UCHKPT_CMD_SAMOG_HANDOFF_INIT_INFO	518
SESS_UCHKPT_CMD_SAMOG_LI_PROV_INFO	519
SESS_UCHKPT_CMD_SAMOG_MIPV6_TIMER_INFO	519
SESS_UCHKPT_CMD_SAMOG_MULTI_ROUND_AUTHEN_INFO	519
SESS_UCHKPT_CMD_SAMOG_REAUTHEN_INFO	520
SESS_UCHKPT_CMD_SAMOG_REAUTHOR_INFO	520
付録 E :	
VPC-SI SDR CLI コマンドストリング	521



## このマニュアルについて

ここでは、『*VPC-SISystem Administration Guide*』、マニュアルの構成、および表記法について説明します。

Cisco 仮想パケット コア シングル インスタンス (VPC-SI) は、既製品 (COTS) のサーバー上の仮想マシン (VM) で実行されている単一の StarOS インスタンスで構成されます。このガイドでは、ハイパーバイザ制御 VM 内で実行されている StarOS インスタンスを設定および管理する方法について説明します。



- (注) コントロールプレーンとユーザプレーンの分離 (CUPS) は、StarOS ベースの製品の 3G、4G、および 5G のネットワークでの展開方法におけるアーキテクチャ上の著しい変更を表します。別段の指定がない限り、このドキュメントで言及されている設定 (コマンド、統計、属性、MIB オブジェクト、アラーム、ログ、サービスを含むがこれらに限定されない) が、レガシー製品または CUPS 以外の製品との機能的な同等性を意味するものではありません。CUPS 製品または機能への言及は、情報提供のみを目的としています。この製品と CUPS 製品間のパリティに関するご質問については、シスコのアカウント担当者またはサポート担当者にお問い合わせください。



- (注) この製品のマニュアルセットは、偏向のない言語を使用するように配慮されています。このドキュメントセットでの偏向のない言語とは、年齢、障害、性別、人種的アイデンティティ、民族的アイデンティティ、性的指向、社会経済的地位、およびインターセクショナリティに基づく差別を意味しない言語として定義されています。製品ソフトウェアのユーザインターフェイスにハードコードされている言語、RFP のドキュメントに基づいて使用されている言語、または参照されているサードパーティ製品で使用されている言語によりドキュメントに例外が存在する場合があります。



- (注) 現在のリリースは、シスコのセキュリティ開発ライフサイクルに完全には準拠していません。シスコは、このリリースに関連するセキュリティの脆弱性を分析して特定し、影響の大きな脆弱性を解決しました。脆弱性は、シスコのセキュリティ脆弱性ポリシーに従って開示されません。

このガイドでは、仮想化プラットフォームで実行されている StarOS を設定および維持する一般的な方法について説明します。また、システムパフォーマンスのモニタリングとトラブルシューティングについても説明します。StarOS の一般的な動作とサポートされているネットワークゲートウェイ機能に関連する補足情報については、StarOS のマニュアルを参照してください。

- [使用する表記法](#) (xxx ページ)
- [関連資料](#) (xxxii ページ)
- [カスタマーサポートへの問い合わせ](#) (xxxii ページ)

## 使用する表記法

次の表に、このマニュアル全体で使用される表記法を示します。

通知タイプ	説明
情報メモ	重要な機能または手順に関する情報を提供します。
注意	プログラム、デバイス、またはシステムに損傷を与えるおそれがあることを注意喚起します。
警告	人身傷害または死亡事故のおそれがあることを警告します。また、電氣的障害のおそれがあることを警告する場合があります。

書体の表記法	説明
スクリーンディスプレイとして表されるテキスト	この書体は、端末画面に表示されるディスプレイを表します。次に例を示します。  ログイン：

書体の表記法	説明
として表現されるテキスト <b>commands</b>	この書体は、入力したコマンドを表します。次に例を示します。  <b>show ip access-list</b>  このマニュアルでは、コマンドの完全表記に常に小文字を使用しています。コマンドには、大文字と小文字の区別はありません。
<b>command</b> 変数として表されるテキスト	この書体は、コマンドの一部である変数を表します。次に例を示します。  <b>show card slot_number</b>  <i>slot_number</i> は、目的のシャーシのスロット番号を表す変数です。
メニュー名またはサブメニュー名として表されるテキスト	この書体は、ソフトウェアアプリケーション内でアクセスするメニューとサブメニューを表します。次に例を示します。  [File] メニュー、[New] の順にクリックしてください。

コマンドシンタックスの表記法	説明
{ <b>keyword</b> または変数 }	必須キーワードオプションと変数は、コマンドシンタックスの一部として入力する必要があるコンポーネントです。  必須キーワードオプションと変数は、波カッコ {} で囲まれています。次に例を示します。  <b>sctp-max-data-chunks { limit max_chunks   mtu-limit }</b>  キーワードまたは変数が、波カッコまたは角カッコで囲まれていない場合、これは必須です。次に例を示します。  <b>snmp trap link-status</b>
[ <b>keyword</b> または変数 ]	オプションのキーワードや変数、またはユーザーが使用するかどうかを選択できるキーワードや変数は、角カッコで囲まれています。

コマンドシンタックスの表記法	説明
	<p>コマンドによっては、複数のオプションをサポートしています。これらは、各オプションを縦棒で区切って、波カッコまたは角カッコ内に記述されます。</p> <p>これらのオプションは、必須キーワードやオプションキーワードまたは変数と組み合わせて使用できます。次に例を示します。</p> <pre>action activate-flow-detection {   intitiation   termination }</pre> <p>または</p> <pre>ip address [ count number_of_packets     size number_of_bytes ]</pre>

## 関連資料

この製品の最新情報は、各ソフトウェアリリースに付属している製品の「リリースノート」で確認できます。

[www.cisco.com](http://www.cisco.com) では、次のユーザードキュメントを入手できます。

- *AAA Interface Administration and Reference*
- *Command Line Interface Reference*
- *GTPP Interface Administration and Reference*
- *IPSec Reference*
- *Release Change Reference*
- *SNMP MIB Reference*
- *Statistics and Counters Reference*
- *Thresholding Configuration Guide*
- 製品固有および機能固有の管理ガイド

## カスタマー サポートへの問い合わせ

カスタマーサポートに連絡するには、この項の情報を使用してください。

最新の製品マニュアルについては、<http://www.cisco.com> のサポートエリアを参照するか、サービスリクエストを送信してください。このサイトにアクセスするには、有効なユーザー名とパスワードが必要です。詳細については、シスコの販売担当者またはサービス担当者にお問い合わせください。





# 第 1 章

## VPC-SI 入門

この章では、シスコの仮想化されたパケットコア（シングルインスタンス（VPC-SI）について説明します。VPC-SI は、モバイル市場の新しいアプリケーションとサービスの迅速な展開を可能にする仮想化されたクラウドアーキテクチャのニーズに対応しています。

- [製品の説明（1 ページ）](#)
- [フィーチャセット（3 ページ）](#)
- [冗長性と可用性（5 ページ）](#)
- [ハイパーバイザ要件（6 ページ）](#)
- [DPDK 内部フォワーダ（8 ページ）](#)
- [キャパシティ、CEPS、およびスループット（11 ページ）](#)
- [診断およびモニタリング（12 ページ）](#)
- [StarOS VPC-SI のビルドコンポーネント（12 ページ）](#)
- [VPC-SI ブートパラメータ（12 ページ）](#)
- [ソフトウェアインストールおよびネットワーク展開（27 ページ）](#)

## 製品の説明

この章では、StarOS VPC-SI アーキテクチャと外部デバイスとの連携動作について説明します。

VPC-SI は、StarOS を実行している物理的な Cisco ASR 5500 シャーシの動作を、市販の（COTS）サーバー上で実行できる 1 つの仮想マシン（VM）に統合します。各 VPC-SI VM は独立した StarOS インスタンスとして動作し、物理シャーシの管理およびセッション処理機能を組み込みます。

コントロールおよびユーザプレーン分離（CUPS）の VNF 展開の UGP では、CF および関連する SF VNF はコントロールプレーン機能（CPF）を構成します。SAEGW サービスは、Sxa、Sxb、および Sxab（Sxa と Sxa の組み合わせ）インターフェイスにコントロールプレーン機能を提供するために使用されます。

CUPS ベースの VNF 展開では、UPF は SAEGW-U として機能します。これは StarOS の 1 つのスタンドアロンインスタンスを実行する VNFC として展開されます。展開要件に基づいて拡張性を得るために、複数の UPF VNFC を展開できます。

Day-N の設定手順については、『*Ultra Packet Core CUPS Control Plane Administration Guide*』と『*Ultra Packet Core CUPS User Plane Administration Guide*』を参照してください。

## 仮想化されたモビリティ機能

VPC-SI は、ネットワークのコア内でモビリティ固有のサービスとアプリケーションを実装する、仮想化されたモビリティ機能のセットで構成されています。これらの機能には、次が含まれています。

- モバイルパケットコア
  - LTE MME (モバイル管理エンティティ)、P-GW (PDN ゲートウェイ)、および S-GW (サービングゲートウェイ)
  - GGSN ゲートウェイ GPRS サポートノード
  - SAE-GW System Architecture Evolution ゲートウェイ
  - SGSN サービング GPRS サポートノード (3G のみ)
- モバイル コントロールプレーン PCRF (ポリシーおよび課金ルール機能)、アプリケーションゲートウェイ、分析、サービス オークストレーション、抽象化および制御機能
- スマールセルゲートウェイ
  - HNBGW ホームノード B ゲートウェイ
  - HeNBGW 進化型ホームノード B ゲートウェイ
  - SAMOG S2a Mobility over GTP は、単一のサービスノードで CGW (統合アクセスゲートウェイ) と信頼された WLAN AAA プロキシ (TWAP) 機能を組み合わせます。

モバイルクラウドネットワーク (MCN) はネットワークインフラストラクチャであり、VPC-SI とその他のサービスが展開される Infrastructure as a Service (IaaS)、オークストレーションメカニズム、分析メカニズムなどが備わっています。

## VM インターコネク ト アーキテクチャ

次の図に、VPC-SI でサポートされている基本的な L2/L3 の相互接続を示します。

上の図では、仮想スイッチはハイパーバイザ内に組み込まれ、データセンター全体の SDN L2 機能をサポートしています。仮想スイッチは 802.1Q トランク (VLAN) を使用して他の仮想スイッチと相互接続します。通常、仮想スイッチは動的にロードされたカーネルモジュールです。

## スタンドアロンインスタンス

VPC-SI は基本的に、COTS プラットフォーム上の仮想マシン (VM) 内で実行されている StarOS です。これは、企業内、リモートサイト内、または顧客のデータセンター内のスタンドアロンの単一 VM として使用できます。また、VPC-SI を大規模なサービスプロバイダー オークストレーション ソリューションの一部として統合することもできます。

シングル インスタンス アーキテクチャは、キャパシティの低いシナリオに適しています。VPC-SI 仮想ネットワーク機能 (VNF) を拡張するには、特定の VNF タイプ (P-GW、S-GW、MME、PCRF、ファイアウォール、NAT など) に対してネットワークレベルを大幅に設定する必要があります。たとえば、新しい VPC-SI P-GW が追加または削除された場合、この情報を使用してさまざまな Diameter ピアを設定する必要があります。この情報によって DNS がプロビジョニングされるか、またはプロビジョニングが解除されます。

VPC-SI は、サポートされているハイパーバイザ KVM (カーネルベースの仮想マシン) と VMware ESXi とのみ連携します。物理的なデバイスの知識はほとんどないか、まったくありません。

通常、VPC-SI はハードウェアまたはハイパーバイザの障害が発生した場合に物理的な冗長性を提供するために、シャーシ間セッションリカバリ (ICSR) ペアに導入する必要があります。

各 VPC-SI VM が StarOS システム全体のロールを担います。VM の外部で公開されている唯一のインターフェイスは、外部管理およびサービストラフィックのインターフェイスです。各 VM は個別に管理されます。

各 VPC-SI VM は次の StarOS 機能を実行します。

- コントローラのタスク
- CLI とログイン用のアウトオブバンド管理 (vSerial と vKVM)
- ローカルコンテキスト vpnmgr
- ローカルコンテキスト管理 (vNIC)
- vHDD のシステム起動イメージと設定ストレージ
- vHDD のレコードストレージ
- fastpath と slowpath を介した NPU シミュレーション
- 非ローカルコンテキスト (vNIC、1~12)。
- セッション処理の逆多重化と vpnmgr
- 暗号処理

## フィーチャセット

### インターフェイスおよびアドレッシング

VM は単一の CPU サブシステムを備えた仮想カードとして表されます。これにより、多くの CLI コマンド、ログ、および機能は、ASR 5500 プラットフォームで実行されている StarOS と同様に動作します。

StarOS 用に作成されたアプリケーションは、単一の仮想カードをサポートする 1 スロットの仮想シャーシを備えた別のプラットフォームとして VPC-SI を見なします。

コンテキスト、サービス、プール、インターフェイス、カード、およびポートの StarOS の概念は既存のプラットフォームの場合と同様に VPC-SI に存在します。

VMが起動すると、VMプロファイル内に設定されたvNICが検出され、「仮想イーサネット」タイプのポートに相当する数が StarOS CLI に表示されます。

- VPC-SIはハイパーバイザによって提供される順序でvNICインターフェイスを割り当てます。
  - 最初に提供されるインターフェイスは、VPC-SI 管理用の 1/1 です。
  - 2番目に提供されるインターフェイスは、VPC-SI サービス制御およびデータトラフィック用の 1/10 です。
  - 任意の 3 番目に提供されるインターフェイスは、VPC-SI サービス制御およびデータトラフィック用の 1/11 です。
  - 任意の 4 番目以降のインターフェイスは、1/12、1/13 になります。
- サポートされているハイパーバイザにリストされているインターフェイスは、VPC-SI インターフェイスと一致させる順序で KVM BR グループまたは VMware vSwitch と一致していることを確認する必要があります。



(注) ハイパーバイザの CLI や GUI にリストされている vNIC の順序が、ハイパーバイザが VPC-SI に提供する方法と同じであることは保証できません。初期設定では、**show hardware** CLI コマンドを使用して、ハイパーバイザのvNIC設定に表示されているMACアドレスを調べ、VPC-SIによって学習されたMACアドレスと一致させる必要があります。これにより、VPC-SIが目的のBRグループまたはvSwitchに接続されていることが確認されます。

## 暗号化

VPC-SI インスタンス内の VM は、ソフトウェアベースの暗号化とパケットのトンネリングを実行します（高いスループットのハードウェアベースのサービスとは異なります）。ベアラーパケットの暗号化を多用するコールモデル、または重要な PKI（公開キーインフラストラクチャ）のキー生成レートを持つコールモデルでは、重要なコンピューティングリソースが必要になる場合があります。

## セキュリティ

外部トラフィックのセキュリティは、トンネリング、暗号化、Access Control List (ACL; アクセスコントロールリスト)、コンテキストの分離、およびユーザー認証機能など、既存の StarOS プラットフォームと同様の機能を備えています。CFおよびSF上のユーザーポートおよびインターフェイスは、StarOS の CLI 設定によって保護されます。

仮想システムでは、データセンター機器の DI ネットワークを介してネットワーク通信が行われるため、お客様側にさらにセキュリティ上の懸念が生じます。

DI ネットワークは、システムネットワークの VLAN のメンバーシップをその特定の VPC-SI インスタンス内の VM に制限することによって、データセンター内の他のホストから分離する必要があります。ネットワークに誤って追加された他のホストによる DI ネットワークへの不正アクセス、またはルータ、スイッチ、またはハイパーバイザの侵害によって、StarOS のセ

セキュリティ対策が中断または回避される可能性があります。このような中断により、障害、サービスの損失や、制御とベアラーパケットの検出が発生する可能性があります。DI ネットワークへのアクセスを適切に保護することは、StarOS の制御の範囲を超えています。

DI ネットワークコンポーネント（CF および SF など）の VM 間の通信は、外部から提供された SSH キーによる認証によってのみ可能になりました。さらに、システムは、DI ネットワーク内のログインに公開/秘密キーベースの SSH 認証を適用します。パスワード、キー、または LI 情報は保存されず、クリアテキストで送信されます。

オペレータが、管理およびベアラーの LI（合法的傍受）などネットワークを物理的に分離する必要がある場合は、センシティブデータを伝送するため、DI ネットワークの物理的な分離も実行する必要があります。仮想環境では、ネットワークの物理的な分離は不可能な場合、または実用的でない場合があります。これらの要件を持つオペレータは、必要に応じて十分な保護を提供することを確認するために、ハイパーバイザとインフラストラクチャを調査する必要があります。

## 冗長性と可用性

### プラットフォーム要件

仮想システムは、リソースの冗長性と可用性全体の基盤となるハードウェアとハイパーバイザに依存しています。StarOS は、その上部でシステムの冗長性を処理します。

ハードウェアとハイパーバイザには、次が備わっている必要があります。

- 実際の冗長ハードウェアコンポーネント（電源、ディスクなど）
- 冗長ネットワークパス（デュアルファブリック/NIC、自動フェールオーバー搭載）
- 冗長ネットワークアップリンク（スイッチ、ルータなど）

基盤となるインフラストラクチャ（ホスト、ハイパーバイザ、およびネットワーク）に期待値を超える可用性と信頼性が備わっている場合にのみ、高可用性を実現できます。システムは、実行されている環境と同じ信頼性のみを備えています。

シャード間セッションリカバリ（ICSR）は、冗長ではないハードウェア（CPU、メモリ、マザーボード、ハイパーバイザソフトウェアなど）に障害が発生した場合に、可用性を高め、リカバリ時間を短縮するためにも推奨されます。ICSR は、ゲートウェイのセッションレベルでのみ冗長性を提供します。

### ICSR のサポート

VPC-SI は、StarOS ソフトウェアリリースで ICSR をサポートするサービスの 2 つのインスタンス間で ICSR をサポートします。複数のサービスタイプが使用されている場合は、ICSR をサポートするサービスのみが ICSR を使用できます。



(注) 2つの VPC-DI インスタンスは、異なる DI ネットワーク上にある必要があります。

ICSR は、サイト/行/ラック/ホストの停止や主要なソフトウェア障害の冗長性をサポートしています。これを行うには、重複しないホストとネットワークインターコネクト上で2つのインスタンスを実行する必要があります。ICSRは、同様に設定されたインスタンス間でのみサポートされます。VPC-SI インスタンスと別のタイプのプラットフォーム間の ICSR はサポートされていません。

L3 ICSR がサポートされています。

詳細については、このガイドの「シャーシ間セッションリカバリ」の章を参照してください。

## IFTASK 冗長性

アクティブスタンバイタスクが強制終了されるか、または停止されると、スタンバイ IFTASK への移行が行われます。この移行プロセスには、スタンバイ IFTASK からアクティブ IFTASK への特定のパラメータの初期化が含まれます。現在のハートビート間隔では、初期化に3秒以上かかる場合があります、カードが再起動します。

カードの再起動を回避するには、次の設定を使用して HAT HB 間隔を高い値に設定します。

### configure

```
[ default ] high-availability fault-detection card hb-loss value
```

## ハイパーバイザ要件

VPC-SI は、次のハイパーバイザで実行する資格があります。

- カーネルベースの仮想マシン (KVM) : QEMU エミュレータ 2.0 VPC-SI StarOS インストールビルドには、Ubuntu Server14.04 下での VM 作成用の libvirt XML テンプレートと `ssi_install.sh` が含まれています。
- KVM : Red Hat Enterprise Linux 7.2 : VPC-SI StarOS インストールビルドには、`qvpc-si_install.sh` というインストールスクリプトが含まれています。
- VMware ESXi 6.0 : VPC-SI StarOS インストールビルドには、ESXi GUI を介した VM の作成用の OVF (オープン仮想化フォーマット) と OVA (オープン仮想アプリケーション) のテンプレートが含まれています。
- VMware ESXi
  - バージョン 6.0 : VPC-SI StarOS インストールビルドには、ESXi GUI を介した VM の作成用の OVF (オープン仮想化形式) および OVA (オープン仮想アプリケーション) テンプレートが含まれています。このバージョンはリリース 21.8 より前のリリースでサポートされています。
  - バージョン 6.5 : リリース 21.8 および 21.9 でサポート

- バージョン 6.7 : リリース 21.10 以降でサポート

## VM の設定

VPC-SI では、VM が次のように設定されている必要があります。

- X vCPU ([vCPU と vRAM のオプション \(7 ページ\)](#) を参照)
- Y vRAM ([vCPU と vRAM のオプション \(7 ページ\)](#) を参照)
- 最初の vNIC は管理ポート ([vNIC のオプション \(8 ページ\)](#) を参照)
- 2 番目以降の vNIC はサービスポートです。1 つの vNIC を必要とし、最大 12 個が VPC でサポートされますが、この数はハイパーバイザによって制限される場合があります。
- 最初の vHDD はブートイメージと設定ストレージ用です (4 GB を推奨)。
- 2 番目の vHDD はレコードストレージ用です (オプション) (16 GB 以上)

## vCPU と vRAM のオプション

CPU は、システム全体とアプリケーションを単独で完全に実行でき、複数の物理 CPU コアの搭載が可能な 1 つの物理コンピュータチップです。仮想コアテクノロジーは、物理コアごとに複数の論理プロセッサ (vCPU) をサポートします。特定のプラットフォーム上でサポートされている vCPU の合計数は、使用可能な物理コアの数と、各コアに実装されている仮想コアテクノロジーのタイプによって異なります。

VM が NUMA ノードにまたがらないようにすることを推奨します。vCPU は基盤となる CPU に合わせます。パフォーマンスを最適化するために、VM が NUMA ノードにまたがらないようにします。numa ノードの数の制限は 2 です。

CF と SF は、多数の vCPU が割り当てられ、それぞれが 1 つのスレッド (命令のシーケンス) VM 内で実行します。プラットフォーム CPU でサポートされている使用可能な vCPU の数がハイパーバイザを介して VM に割り当てることができる vCPU の最大数を超える場合があります。



- (注) VM あたりの vCPU の数は、プラットフォーム CPU でサポートされている vCPU の最大数を超えてはなりません。

パフォーマンスを最大化するには、基盤となるハードウェアに合わせて、vCPU または vRAM の数を調整することが望ましい場合があります。SF は vCPU と vRAM のさまざまな組み合わせをサポートしますが、すべての SF がインスタンス内で同じ組み合わせを共有する必要があります。

ソフトウェアは、vCPU の数とその SF の vRAM の量に基づいて、SF の起動時に SF ごとに最適な SESSMGR タスクの数を決定します。



(注) vCPU の数、vRAM のサイズ、または vNIC のタイプ/カウント（ホットプラグ、バルーニングなどを介す）の動的なサイズ変更はサポートされていません。プロビジョニング後にこれらの値を変更する必要がある場合は、すべての VM をシャットダウンして再設定する必要があります。CPU と RAM が他のインスタンスと一致しなくなるため、一度に 1 つの VM を再設定することはできません。

## vNIC のオプション

このリリースでは、次のような vNIC オプションがサポートされています。

- VMXNET3 : VMware 用の準仮想 NIC
- VIRTIO : KVM 用の準仮想 NIC
- ixgbe : Intel 10 ギガビット NIC 仮想機能
- enic : Cisco UCS NIC

## ハードドライブストレージ

必須/フラッシュ（非 RAID）ドライブに追加した場合、システムは仮想マシン（VM）の下で RAID1 をサポートします。VM ごとに、この表に示す SCSI ID と一致する仮想 SCSI ディスクを CF のみで作成できます。最小ディスクサイズは 16 GB を超えている必要があります。

表 1: ディスクマッピング

タイプ	/flash (非 RAID)	hd-local1	注意
KVM	SCSI 0:0:0:0	SCSI 0:0:1:0	Raw ディスクの hd-local1 は RAID1 を使用
VMware	SCSI 0:0:0:0	SCSI 0:0:1:0	Raw ディスクの hd-local1 と hd-remote1 は R します

記録ストレージ（CDR および UDR）の場合、CF VM には、予想される記録要件を満たすサイズの 2 番目の vHDD をプロビジョニングする必要があります（最小 16 GB）。レコードは、2 番目の vHDD 上の /records に書き込まれます。

## DPDK 内部フォワーダ

Intel Data Plane Development Kit (DPDK) は、VPC-SI アーキテクチャの不可欠な部分であり、8 個以上の vCPU で設定された VPC-SI システムのパフォーマンスを向上させるために使用されます。DPDK 内部フォワーダ (IFTASK) は、パケットの入出力操作を担当するソフトウェ



アコンポーネントであり、Linux カーネルをバイパスすることによって、ユーザー空間でのパケット処理の高速パスを提供します。VPC-SI ブートプロセス中に、vCPU の比率が IFTASK に割り当てられ、残りの部分はアプリケーション処理に割り当てられます。

IFTASK で使用される vCPU を特定し、それらの使用率を表示するには、**show npu utilization table** コマンドを次のように使用します。

```
[local]mySystem# show npu utilization table

***** show npu utilization table card 1 *****
          5-Sec Avg:
lcore00|lcore01|lcore02|lcore03|lcore04|lcore05|lcore06|lcore07|
          IDLE:           |    4%|    43%|    24%|    45%|    17%|    |
          |
          QUEUE_PORT_RX:  |   34%|   56%|    |    |    |    |
          |
          QUEUE_PORT_TX:  |    |    |    |    |    |    |
          |
          QUEUE_VNPU_RX:  |    |    |    |    |    |    |
          |
          QUEUE_VNPU_TX:  |    |    |    |    |    |    |
          |
          QUEUE_KNI_RX:   |   57%|    |    |    |    |    |
          |
          QUEUE_KNI_TX:   |    |    |    |    |    |    |
          |
          QUEUE_THREAD_KNI: |   3%|    |    |    |    |    |
          |
          QUEUE_MCDMA_RX: |    |    |    |   7%|    |   5%|    |
          |
          QUEUE_MCDMA_TX: |    |    |    |    |    |    |
          |
          QUEUE_THREAD_MCDMA: |    |    |    |   8%|    |  24%|    |
          |
          QUEUE_THREAD_VNPU: |    |    |    |    |    |    |
          |
          QUEUE_CRYPTO_RX: |    |    |    |    |    |    |
          |
          QUEUE_CRYPTO_TX: |    |    |    |    |    |    |
          |
          QUEUE_THREAD_IPC: |    |    |    |    |    |    |
          |
          MCDMA_FLUSH:    |    |    |    |  59%|  54%|  51%|    |
          |
          QUEUE_THREAD_TYPE_MAX: |    |    |    |    |    |    |
          |
          300-Sec Avg:
lcore00|lcore01|lcore02|lcore03|lcore04|lcore05|lcore06|lcore07|
          IDLE:           |   99%|  100%|   31%|   30%|   32%|    |
          |
          QUEUE_PORT_RX:  |   0%|    |    |    |    |    |
          |
          QUEUE_PORT_TX:  |    |    |    |    |    |    |
          |
          QUEUE_VNPU_RX:  |    |    |    |    |    |    |
          |
          QUEUE_VNPU_TX:  |    |    |    |    |    |    |
          |
          QUEUE_KNI_RX:   |    |    |    |    |    |    |
          |
          QUEUE_KNI_TX:   |    |    |    |    |    |    |
          |
          QUEUE_THREAD_KNI: |    |    |    |    |    |    |
```

```

|
|   QUEUE_MCDMA_RX:          |   |   |   |   0%|   0%|   0%|   |
|   QUEUE_MCDMA_TX:          |   |   |   |   |   |   |   |
|   QUEUE_THREAD_MCDMA:      |   |   |   |   |   |   |   |
|   QUEUE_THREAD_VNPU:       |   |   |   |   |   |   |   |
|   QUEUE_CRYPTO_RX:         |   |   |   |   |   |   |   |
|   QUEUE_CRYPTO_TX:         |   |   |   |   |   |   |   |
|   QUEUE_THREAD_IPC:        |   |   |   |   |   |   |   |
|   MCDMA_FLUSH:             |   |   |   | 68%| 69%| 67%|   |
|   QUEUE_THREAD_TYPE_MAX:    |   |   |   |   |   |   |   |
|
|   900-Sec Avg:
|   lcore00|lcore01|lcore02|lcore03|lcore04|lcore05|lcore06|lcore07|
|   IDLE:          | 99%| 100%| 31%| 31%| 32%|   |
|
|   QUEUE_PORT_RX:         | 0%|   |   |   |   |   |   |
|   QUEUE_PORT_TX:         |   |   |   |   |   |   |   |
|   QUEUE_VNPU_RX:         |   |   |   |   |   |   |   |
|   QUEUE_VNPU_TX:         |   |   |   |   |   |   |   |
|   QUEUE_KNI_RX:          |   |   |   |   |   |   |   |
|   QUEUE_KNI_TX:          |   |   |   |   |   |   |   |
|   QUEUE_THREAD_KNI:      |   |   |   |   |   |   |   |
|   QUEUE_MCDMA_RX:        |   |   |   | 0%| 0%| 0%|   |
|   QUEUE_MCDMA_TX:        |   |   |   |   |   |   |   |
|   QUEUE_THREAD_MCDMA:    |   |   |   |   |   |   |   |
|   QUEUE_THREAD_VNPU:     |   |   |   |   |   |   |   |
|   QUEUE_CRYPTO_RX:       |   |   |   |   |   |   |   |
|   QUEUE_CRYPTO_TX:       |   |   |   |   |   |   |   |
|   QUEUE_THREAD_IPC:      |   |   |   |   |   |   |   |
|   MCDMA_FLUSH:           |   |   |   | 68%| 68%| 67%|   |
|   QUEUE_THREAD_TYPE_MAX:  |   |   |   |   |   |   |   |
|

```

```

thread 2 QUEUE_PORT_RX          77.22 %
thread 2 IDLE                    22.78 %
-----
thread 5 MCDMA_FLUSH            57.74 %
thread 5 IDLE                    40.13 %
thread 5 QUEUE_THREAD_MCDMA     2.13 %

```

```

-----
thread 1 QUEUE_KNI_RX                50.39 %
thread 1 QUEUE_PORT_RX               40.72 %
thread 1 IDLE                         6.13 %
thread 1 QUEUE_THREAD_KNI            2.76 %
-----
thread 3 QUEUE_THREAD_MCDMA          41.17 %
thread 3 MCDMA_FLUSH                 38.31 %
thread 3 IDLE                        16.28 %
thread 3 QUEUE_MCDMA_RX              4.24 %
-----
thread 4 IDLE                         56.03 %
thread 4 MCDMA_FLUSH                 43.97 %
-----

```

IFTASK vCPU を使用せずに VM の CPU 使用率を表示するには、**show cpu info** コマンドを使用します。詳細については、**verbose** キーワードを使用してください。

```

[local]mySystem# show cpu info
Card 1, CPU 0:
  Status                : Active, Kernel Running, Tasks Running
  Load Average          : 8.99, 9.50, 8.20 (11.89 max)
  Total Memory           : 16384M
  Kernel Uptime         : 0D 0H 49M
  Last Reading:
    CPU Usage           : 16.6% user, 10.5% sys, 0.0% io, 4.6% irq, 68.3% idle
    Poll CPUs           : 5 (1, 2, 3, 4, 5)
    Processes / Tasks   : 234 processes / 54 tasks
    Network              : 353.452 kpps rx, 3612.279 mbps rx, 282.869 kpps tx, 2632.760
    mbps tx
    File Usage           : 2336 open files, 1631523 available
    Memory Usage         : 4280M 26.1% used, 42M 0.3% reclaimable
  Maximum/Minimum:
    CPU Usage           : 23.2% user, 11.2% sys, 0.1% io, 5.5% irq, 61.5% idle
    Poll CPUs           : 5 (1, 2, 3, 4, 5)
    Processes / Tasks   : 244 processes / 54 tasks
    Network              : 453.449 kpps rx, 4635.918 mbps rx, 368.252 kpps tx, 3483.816
    mbps tx
    File Usage           : 3104 open files, 1630755 available
    Memory Usage         : 4318M 26.4% used, 46M 0.3% reclaimable

```

## キャパシティ、CEPS、およびスループット

VPC-SI インスタンスのサイジングには、予想されるコールモデルのモデリングが必要です。

サービスタイプの多くは、他よりも多くのリソースを必要としています。パケットサイズ、セッションあたりのスループット、CEPS（コールイベント/秒）レート、IPSecの使用率（サイト間、サブスクライバ、LI）、その他のVMとの競合、および基盤となるハードウェアのタイプ（CPUの速度、vCPUの数）によって、ブスクライバの最大有効数がさらに制限されます。同等のハードウェアとハイパーバイザ設定でのコールモデルの認定が必要です。

## 診断およびモニタリング

VPC-SIはVM内で実行されるため、ハードウェアの診断またはモニタリングは行われません。ハードウェアセンサーデータ（温度、電圧、メモリエラー）の取得は、ハイパーバイザと外部モニタリングシステムを介して行われます。

VPC-SIは、CLI **show** コマンド、バルク統計情報、およびMIBトラップなどの既存のメカニズムを通じてvCPU、vRAM、およびvNICの使用状況をモニターし、エクスポートします。ただし、オペレータはハイパーバイザ内の物理CPU、RAM、およびNICの値のモニタリングのほうが有用であると気づく場合があります。

vNICには可変の最大スループット（たとえば、1 Gbps または 10 Gbps として定義されていない）があるため、スループットのパーセンテージとして使用率をエクスポートするカウンタとバルク統計情報の値がほとんどない場合があります。絶対値（bps）はVPC-SIから取得できませんが、物理インフラストラクチャの使用率をハイパーバイザから取得する必要があります。これは、最大スループットが固定されるため、パススルー PF NICには適用されません。

## StarOS VPC-SI のビルドコンポーネント

次の StarOS ビルドのファイル名のタイプは、次のように VPC-SI と関連付けられています。

- **qvmc-si-<version>.iso** : 初期インストールまたはスタートオーバー ISO ファイル
- **qvmc-si-<version>.bin** : すでに実行されているシステムの更新、アップグレード、またはリカバリファイル。詳細については、「*StarOS* 管理操作」の章を参照してください。
- **qvmc-si-template-libvirt-kvm-<version>.tgz** : KVM libvirt テンプレートと `ssi_install.sh`
- **qvmc-si-template-vmware-<version>.ova** : VMware OVA テンプレート
- **qvmc-si-<version>.qcow2.gz** : KVM QCOW2 ディスクテンプレート

## VPC-SI ブートパラメータ

起動パラメータファイルは、起動する前に StarOS に設定項目を渡すための手段を提供します。これらのパラメータは、管理、サービス、および VNFM インターフェイスの詳細などの項目と StarOS の起動時に作成された内部フォワーダタスク（iftask）の設定を指定します。

起動パラメータは複数の方法で送信され、すべての方式が同じパラメータ名と使用方法を使用します。ブートパラメータファイルの最初の場所は、最初の VM ドライブ（`boot1/param.cfg` など）の最初のパーティションになります。検索された 2 番目の場所は、仮想 CD-ROM ドライブである設定ドライブ上にあります。OpenStack を使用している場合は、ターゲットの起動パラメータのファイル名を `staros_param.cfg` として指定します。OpenStack を使用していない場合は、ルートディレクトリに `staros_param.cfg` を使用して ISO イメージを作成し、この ISO を VM の最初の仮想 CD-ROM ドライブに接続します。

VMが起動すると、最初にCFEと呼ばれるブート前環境でparam.cfgファイルが解析されます。VMがLinuxを開始すると、仮想CD-ROMドライブにアクセスして、staros\_param.cfgファイルを解析します。/boot1/param.cfgファイルに保存されている値との競合がある場合は、staros\_param.cfgのパラメータが優先されます。

## 起動パラメータファイルのフォーマット

起動パラメータファイルの構造は次のとおりです。

```
VARIABLE_NAME = VALUE
```

行区切りとして改行を使用して、1行につき1つの変数を指定します（UNIXテキストファイル形式）。変数名と値は大文字と小文字が区別されません。無効な値は無視され、エラー通知がVMコンソールに表示されます。変数の値が重複している場合（同じ変数名に2つの異なる値が指定されている場合）、最後に定義された値が使用されます。

数値にゼロを埋める必要はありません。たとえば、PCI\_IDは0:1:1.0の場合、0000:01:01.0と同様に扱われます。

## ネットワーク インターフェイス ID

VPC-SIハイパーバイザによって提供される順序でvNICインターフェイスを割り当てます。ハイパーバイザのCLIやGUIにリストされているvNICの順序が、ハイパーバイザがVMに提供する方法と同じであることは保証できません。

VPC-SIがvNICを検出する順序はPCIバスの列挙順に従い、準仮想デバイスもPCIバスで表されます。PCIバスは、同じレベルの追加デバイスの前にブリッジが探索される深さ優先の方法で列挙されます。すべてのネットワークインターフェイスのタイプが同じ場合、PCIトポロジを認識するだけでvNICの正しい順序を取得できます。ネットワークインターフェイスのタイプが異なる場合、その順序はPCIトポロジに加えてVM内のデバイスドライバのロード順序に依存します。デバイスドライバのロード順序は、ソフトウェアのリリース順と同じである保証はありませんが、一般的には準仮想デバイスがパススルーデバイスよりも優先されます。

NICを識別するために使用できる方式はいくつかあります。

- MACアドレス：インターフェイスのMACアドレス
- 仮想PCI ID
- 結合インターフェイス：ネットワークデバイスのボンディングを使用すると、ネットワークインターフェイスはスレーブインターフェイスロールとして機能するように識別されます。ボンドのスレーブインターフェイスは、MAC、PCIID、またはインターフェイスタイプを使用して識別されます。
- インターフェイスタイプおよびインスタンス番号

## 仮想 PCI ID

PCI バス上のデバイスは、ドメイン、バス、デバイス、および機能番号と呼ばれる一意のタプルによって識別されます。これらの識別子は、いくつかの方法で識別できます。

ゲスト内では、**lspci** ユーティリティによって次のようにバスの設定が表示されます。

```
# lspci
```

```
00:00.0 Host bridge: Intel Corporation 440FX - 82441FX PMC [Natoma] (rev 02)
00:01.0 ISA bridge: Intel Corporation 82371SB PIIX3 ISA [Natoma/Triton II]
00:01.1 IDE interface: Intel Corporation 82371SB PIIX3 IDE [Natoma/Triton II]
00:01.2 USB controller: Intel Corporation 82371SB PIIX3 USB [Natoma/Triton II] (rev 01)
00:01.3 Bridge: Intel Corporation 82371AB/EB/MB PIIX4 ACPI (rev 03)
00:02.0 VGA compatible controller: Cirrus Logic GD 5446
00:03.0 System peripheral: Intel Corporation 6300ESB Watchdog Timer
00:04.0 Unclassified device [00ff]: Red Hat, Inc Virtio memory balloon
00:05.0 Ethernet controller: Red Hat, Inc Virtio network device
00:06.0 Ethernet controller: Red Hat, Inc Virtio network device
```

この仮想バスのドメイン、バス、デバイス、および機能番号を次に示します。

表 2: 仮想 PCI ID

回線	ドメイン	バス	デバイス	機能
00:00.0 Host bridge: Intel Corporation 440FX - 82441FX PMC [Natoma] (rev 02)	0	0	0	0
00:01.0 ISA bridge: Intel Corporation 82371SB PIIX3 ISA [Natoma/Triton II]	0	0	1	0
00:01.1 IDE インターフェイス : Intel Corporation 82371SB PIIX3 IDE [Natoma/Triton II]	0	0	1	1
00:01.2 USB コントローラ : Intel Corporation 82371SB PIIX3 USB [Natoma/Triton II] (rev 01)	0	0	1	2
00:01.3 ブリッジ : Intel Corporation 82371AB/EB/MB PIIX4 ACPI (rev 03)	0	0	1	3
00:02.0 VGA 互換コントローラ : Cirrus Logic GD 5446	0	0	2	0
00:03.0 システム周辺機器 : Intel Corporation 6300ESB ウォッチドッグタイマー	0	0	3	0
00:04.0 未分類のデバイス [00ff] : Red Hat, Inc Virtio メモリバルーン	0	0	4	0
00:05.0 イーサネットコントローラ : Red Hat, Inc Virtio ネットワークデバイス	0	0	5	0

回線	ドメイン	バス	デバイス	機能
00:06.0 イーサネットコントローラ : Red Hat, Inc Virtio ネットワークデバイス	0	0	6	0

libvirt ベースの仮想マシンの場合は、**virsh dumpxml** コマンドから仮想 PCI バストポロジを取得できます。libvirt スキーマでは、デバイス番号に *slot* という用語が使用されることに注意してください。これは、前の例で使用した仮想マシンにおける xml の説明のスニペットです。

```
<interface type='bridge'>
  <mac address='52:54:00:c2:d0:5f'>/>
  <source bridge='br3043'>/>
  <target dev='vnet0'>/>
  <model type='virtio'>/>
  <driver name='vhost' queues='8'>/>
  <alias name='net0'>/>
  <address type='pci' domain='0x0000' bus='0x00' slot='0x05' function='0x0'>/>
</interface>
<interface type='bridge'>
  <mac address='52:54:00:c3:60:eb'>/>
  <source bridge='br0'>/>
  <target dev='vnet1'>/>
  <model type='virtio'>/>
  <alias name='net1'>/>
  <address type='pci' domain='0x0000' bus='0x00' slot='0x06' function='0x0'>/>
</interface>
```

### インターフェイスタイプおよびインスタンス番号

ここで NIC は、Linux デバイスドライバの名前 (virtio\_net、vmxnet3、ixgbe、i40e など) とそのインスタンス番号を使用して、そのタイプによって識別されます。インスタンス番号は、そのタイプのインターフェイスの PCI 列挙順に基づいています。インスタンス番号 1 から始まります。インターフェイスタイプは、パススルーインターフェイスと SR-IOV 仮想機能だけでなく、両方の準仮想タイプを識別するために使用できます。PCI バス上のデバイスの PCI 列挙の順序は、ホスト OS 上の **lspci** ユーティリティで確認できます。

たとえば、次のゲスト PCI トポロジの CF は、virtio\_net インターフェイス番号 1 が 00:05.0 のイーサネットコントローラであり、virtio\_net インターフェイス番号 2 が 00:06.0 のイーサネットコントローラであることを示しています。出力は、ゲストで実行された **lspci** コマンドからのものです。

```
# lspci
```

```
00:00.0 Host bridge: Intel Corporation 440FX - 82441FX PMC [Natoma] (rev 02)
00:01.0 ISA bridge: Intel Corporation 82371SB PIIX3 ISA [Natoma/Triton II]
00:01.1 IDE interface: Intel Corporation 82371SB PIIX3 IDE [Natoma/Triton II]
00:01.2 USB controller: Intel Corporation 82371SB PIIX3 USB [Natoma/Triton II] (rev 01)
00:01.3 Bridge: Intel Corporation 82371AB/EB/MB PIIX4 ACPI (rev 03)
00:02.0 VGA compatible controller: Cirrus Logic GD 5446
00:03.0 System peripheral: Intel Corporation 6300ESB Watchdog Timer
00:04.0 Unclassified device [00ff]: Red Hat, Inc Virtio memory balloon
00:05.0 Ethernet controller: Red Hat, Inc Virtio network device
```

00:06.0 Ethernet controller: Red Hat, Inc Virtio network device

サポートされている Linux ドライバの完全なリストを次に示します。

表 3: サポートされている Linux ドライバ

タイプ	PCI ベンダー/デバイス ID	ドライバ名
VIRTIO (KVM 用の準仮想 NIC)	0x10af / 0x1000	virtio_net
VMXNET3 (VMware 用の準仮想 NIC)	0x15ad / 0x07b0	vmxnet3
Intel 10 ギガビットイーサネット	0x8086 / 0x10b6 0x8086 / 0x10c6 0x8086 / 0x10c7 0x8086 / 0x10c8 0x8086 / 0x150b 0x8086 / 0x10dd 0x8086 / 0x10ec 0x8086 / 0x10f1 0x8086 / 0x10e1 0x8086 / 0x10db 0x8086 / 0x1508 0x8086 / 0x10f7 0x8086 / 0x10fc 0x8086 / 0x1517 0x8086 / 0x10fb 0x8086 / 0x1507 0x8086 / 0x1514 0x8086 / 0x10f9 0x8086 / 0x152a 0x8086 / 0x1529 0x8086 / 0x151c 0x8086 / 0x10f8 0x8086 / 0x1528 0x8086 / 0x154d 0x8086 / 0x154f 0x8086 / 0x1557	ixgbe



タイプ	PCI ベンダー/デバイス ID	ドライバ名
Intel 10 ギガビット NIC 仮想機能	0x8086 / 0x10ed 0x8086 / 0x1515	ixgbevf
Cisco UCS NIC	0x1137 / 0x0043 0x1137 / 0x0044 0x1137 / 0x0071	enic
Mellanox ConnectX-5  (注) <ul style="list-style-type: none"> <li>• Mellanox は、ユーザープレーンでサポートされています。</li> <li>• Mellanox は、VPC-DI の管理インターフェイスでサポートされています。</li> </ul>	0x15b3 / 0x1017 0x15b3 / 0x1018	mlx5_core
Intel XL 710 ファミリ NIC (PF)	0x8086 / 0x1572 (40 ギガ) 0x8086 / 0x1574 (40 ギガ) 0x8086 / 0x1580 (40 ギガ) 0x8086 / 0x1581 (40 ギガ) 0x8086 / 0x1583 (40 ギガ) 0x8086 / 0x1584 (40 ギガ) 0x8086 / 0x1585 (40 ギガ) 0x8086 / 0x158a (25 ギガ) 0x8086 / 0x158b (25 ギガ)	i40e**
Intel XL 710 ファミリ NIC 仮想機能	0x8086 / 0x154c	i40evf

\*\* 注 : i40e ドライバの使用時に、ホスト上で作成された SRIOV VF に対して MAC アドレスの割り当てが動的に行われないという既知の問題が存在します。StarOS VM を起動するには、MAC アドレスの割り当てが必要です。回避策として、MAC アドレスの割り当てはホストから設定する必要があります。詳細は次のリンクを参照してください。 <https://www.intel.com/content/dam/www/public/us/en/documents/technology-briefs/xl710-sr-io-v-config-guide-gbe-linux-brief.pdf>

## VPC-SI ブートパラメータの設定

VPC-SI VM には、管理ネットワークに接続するために 1 つのインターフェイスが設定されています。このインターフェイスは通常、StarOS で設定され、Day 0 設定に含まれている必要があります。管理インターフェイスは、メインの StarOS 設定ファイルを使用してスタティックアドレスの割り当てをサポートします。

0 ~ 4 個の追加ネットワーク インターフェイスはサービスポートとして機能します。これらのインターフェイスは、StarOS によって設定されます。通常、これらのポートは VNF インフラストラクチャ (VNFI) のトランクポートとして設定されます。

仮想ネットワーク機能 (VNF) マネージャ (VNFМ) が存在する場合は、それに接続されているネットワーク インターフェイスを設定するオプションが VPC-SI VM に備わっています。このインターフェイスは、DHCP またはスタティック IP 割り当てを使用して設定でき、VNFМ 以上のレベルのオーケストレータとの通信に使用されます。このインターフェイスは、メインアプリケーションが起動する前に有効になります。

表 4: VPC-SI ブートパラメータ

パラメータ	説明
<b>MGMT_INTERFACE</b> =value	<p>VPC-SI VM 上の管理ポートへのインターフェイス有効な値は次のとおりです。</p> <ul style="list-style-type: none"> <li>• MAC: xx:xx:xx:xx:xx:xx</li> <li>• PCI_ID: xxx:xx:xx.x (Domain:Bus:Device.Function)</li> <li>• TYPE : drive-name-instance-number</li> <li>• BOND : slave-interface-A,slave-interface-B</li> </ul> <p>インターフェイス識別子の決定については、<a href="#">ネットワーク インターフェイス ID (13 ページ)</a> を参照してください。</p>
<b>SERVICE#_INTERFACE</b> =value	<p>サービスポート番号 # へのインターフェイス (# は 1 ~ 4)。</p> <p>サービスインターフェイスは、<b>MGMT_INTERFACE</b> についての上記と同じ値をサポートしています。</p>
<b>SERVICE_INTERFACE_MTU</b> =value	<p>デフォルトでは、IFTASK プロセスは最大インターフェイス MTU を 2,100 バイトに設定します。</p> <p>value は 576 ~ 9,100 バイトの整数にする必要があります。</p>

パラメータ	説明
<b>VNFM_INTERFACE</b> = <i>value</i>	仮想ネットワーク機能マネージャ (VNFM) またはオーケストレータへのオプションのネットワークインターフェイス。  VNFM インターフェイスでは、 <b>MGMT_INTERFACE</b> についての上記と同じ値がサポートされています。
<b>VNFM_IPV4_ENABLE</b> ={ <i>true</i>   <i>false</i> }	VNFM インターフェイスを有効にします。
<b>VNFM_IPV4_DHCP_ENABLE</b> ={ <i>true</i>   <i>false</i> }	VNFM に対して DHCP を有効にします。
<b>VNFM_IPV4_ADDRESS</b> = <i>value</i>	DHCP が使用されていない VNFM の IP アドレスを指定します。
<b>VNFM_IPV4_NETMASK</b> = <i>value</i>	DHCP が使用されていない VNFM の IP アドレスのネットマスクを指定します。
<b>VNFM_IPV4_GATEWAY</b> = <i>value</i>	DHCP が使用されていない VNFM の IP アドレスのゲートウェイを指定します。
<b>IFTASK_CRYPTCO_CORES</b> = <i>value</i>	(任意) (任意) IFTASK_CRYPTCO_CORES IFTASK_SERVICE_TYPE が「2」 (EPDG) に設定されている場合、このパラメータは <i>iftask</i> コアの暗号処理に割り当てるパーセンテージを指定します。値の範囲は 0～50パーセントですが、専用のコアの上限は 4 です。 デフォルトは 0 です。 このパラメータは IFTASK_SERVICE_TYPE が「2」 (EPDG) に設定されている場合にのみ使用する必要があります。他のサービスタイプに設定されている場合、このパラメータは「0」に設定する必要があります。

パラメータ	説明
<b>IFTASK_MCDMA_CORES=value</b>	<p>(任意) マルチチャネルダイレクトメモリアクセス (MCDMA) に割り当てられたコア数を iftask コアの総数の割合として設定します。まず、上記の IFTASK_CORES パラメータを定義する必要があります。注: NUMA 最適化が有効になっており、この MCDMA コアカウント設定も構成されている場合は、IFTASK_CORES=percentage-of-cores を偶数に設定する必要があります。これにより、MCDMA スレッドが NUMA ノード間で均等に分散されます。</p>
<b>CHASSIS_ID=value</b>	<p>シャーシ ID は、CLI 設定 (管理者クレデンシャル、SNMP コミュニティ、RADIUS/Diameter データ、認証データなど) で select コマンドを保護します。設定を保存またはロードする必要があります。通常は、CLI コマンドからランダムなデータとユーザーシードを使用して生成されます。その後、シャーシ ID はシステムの「/flash」ストレージに保存され、後で保存された設定をロードするために使用されます。</p> <p>このオプションを設定すると、/flash のパラメータコンフィギュレーションファイルからシャーシ ID がロードされます。/flash のシャーシ ID は無視され (以前のバージョンからのアップグレードをサポート)、CLI コマンドを使用した新しいシャーシ ID の生成は禁止されます。この方法を使用して指定した場合、シャーシ ID は /flash に保存されなくなります。また、この操作では、ESC (または別の VNFМ) で VM を管理し、シャーシ ID を指定する必要もあります。</p>

パラメータ	説明
<b>IFTASK_SERVICE_TYPE=value</b>	<p>(任意) サービスメモリを計算しサービス固有の機能を有効にするために展開するサービスタイプを指定します。デフォルトは0です。</p> <p>次のサービスタイプを指定できます。</p> <ul style="list-style-type: none"> <li>• 0 = VPC サービスタイプ</li> <li>• 1 = GiLAN サービスタイプ</li> <li>• 2 = ePDG サービスタイプ</li> <li>• 3 = CUPS コントローラサービスタイプ</li> <li>• 4 = CUPS フォワーダサービスタイプ</li> </ul>
<b>FORWARDER_TYPE=value</b>	<p>フォワーダタイプを「vpp」または「iftask」として指定します。</p> <p>デフォルトでは、フォワーダタイプは「iftask」です。フォワーダタイプは、<b>FORWARDER_TYPE</b> が <b>VPP</b> として適切に設定されている場合にのみ、<b>VPP</b> になります。</p> <p>例：<b>FORWARDER_TYPE=vpp</b></p>

#### params.cfg の例

```

MGMT_INTERFACE=MAC:00:01:02:03:04:05
SERVICE1_INTERFACE=PCI_ID:0000:01:02.0
SERVICE2_INTERFACE=PCI_ID:0000:01:03.0
VNFM_INTERFACE=MAC:00:01:02:03:04:AA
VNFM_IPV4_ENABLE=true
VNFM_IPV4_DHCP_ENABLE=false
VNFM_IPV4_ADDRESS=10.1.1.100
VNFM_IPV4_NETMASK=255.255.255.0
VNFM_IPV4_GATEWAY=10.1.1.1
IFTASK_MCDMA_CORES=20
CHASSIS_ID=2sg9x1wqbj
IFTASK_SERVICE_TYPE=0

```

StarOS コマンド **show cloud hardware iftask** を使用して iftask ブートパラメータが有効になったことを確認します。

## VNFM インターフェイスオプションの設定



(注) これらの設定オプションは任意です。

仮想ネットワーク機能マネージャ（VNFM）インターフェイスは、各 VM と VNFM の間で通信するように設計されています。VNFM インターフェイスはメインアプリケーションの前に初期化するため、インターフェイスを設定できるのは起動パラメータのみとなります。

デフォルトでは、VNFM インターフェイスは無効になっています。

#### VNFM IPv4 インターフェイスの有効化

デフォルト値は [False]（無効）です。

変数	有効な値
VNFM_IPV4_ENABLE	True または False

#### IPv4 DHCP クライアントの設定

変数	有効な値
VNFM_IPV4_DHCP_ENABLE	True または False

#### IPv4 スタティック IP の設定



(注) IPv4 DHCP クライアントが有効になっている場合、スタティック設定パラメータは無視されません。

変数	有効な値
VNFM_IPV4_ADDRESS	x.x.x.x
VNFM_IPV4_NETMASK	x.x.x.x
VNFM_IPV4_GATEWAY	x.x.x.x

VNFM IPv6 インターフェイスを有効にします。

変数	有効な値
VNFM_IPV6_ENABLE	True または False

#### IPv6 スタティック IP 設定の有効化

変数	有効な値
VNFM_IPV6_STATIC_ENABLE	True または False

True に設定すると、次の項に示すように、スタティック IP パラメータの設定がインターフェイスに適用されます。False に設定すると、インターフェイスはステートレス自動設定 (RFC4862) と DHCPv6 の両方を使用してインターフェイスのアドレスを設定しようとしません。

### IPv6 スタティック IP の設定



- (注) 「VNFM\_IPV6\_ENABLE」パラメータ値が false に設定されている場合、スタティック設定パラメータは無視されます。IPv6 アドレスフィールドは、RFC 5952 に準拠している必要があります。プレフィックスは /64 で固定されています。

変数	有効な値
VNFM_IPV6_ADDRESS	X:X:X:X:X:X
VNFM_IPV6_GATEWAY	X:X:X:X:X:X

## VPP 設定パラメータ

以降の項では、FORWARDER\_TYPE に VPP を選択した場合にのみ適用されるパラメータを示します。これらのパラメータを使用すると、VPP の CPU とインターフェイス設定をきめ細かく制御できます。



- (注) どのような VPP 設定パラメータをオーバーライドする場合も、事前にシスコのアカウント担当者に連絡してオーバーライド値を確認してください。

### VPP の CPU 割り当て

VPP ワーカーは、CPU コア全体を消費するリアルタイムのスレッドです。VPP のメインスレッドはコア全体を消費しませんが、ビジー状態になる可能性があります。したがって、競合を回避するために割り当てます。

次の表に、VPP-CPU パラメータを示します。

表 5: VPP-CPU パラメータ

パラメータ	説明
<b>VPP_CPU_MAIN=value</b>	<p>Linux プロセッサ番号を指定します。これは 0 ~ (CPU の数から 1 を差し引いた数) になります。デフォルト値は 1 です。</p> <p>次の例を使用して、メインスレッドの値を Linux プロセッサ番号 1 に設定します。</p> <pre>VPP_CPU_MAIN=1</pre>
<b>VPP_CPU_WORKER_CNT=value</b>	<p>Linux プロセッサに設定されているワーカー スレッドの数を指定します。有効な値は 0 ~ (CPU の数から 3 を差し引いた数) です。デフォルト値は、Linux CPU の 50 %、または VPP_CPU_WORKER_LIST 内の CPU の数です。</p> <p>ワーカー スレッドの数を 3 に設定するには、次の例を使用します。</p> <pre>VPP_CPU_WORKER_CNT=3</pre>
<b>VPP_CPU_WORKER_LIST=value</b>	<p>Linux プロセッサに設定されたワーカー スレッドを指定します。ワーカー リストは、Linux プロセッサ番号をカンマで区切ったリストです。有効な値は 0 ~ (CPU の数から 1 を差し引いた数) です。デフォルトは、すべてのソケットに割り当てられるラウンドロビン番号であり、すべてのソケットの最初のコアと最初のソケットの 2 番目のコアをスキップします。</p> <p>次の例を使用して、ワーカー スレッドの数を Linux プロセッサ 数 2、9、および 10 に設定します。</p> <pre>VPP_CPU_WORKER_LIST=2,9,10</pre>

### デフォルトの DPDK 設定

表に示す次のパラメータは、一般的に DPDK を設定するか、またはインターフェイスのデフォルトを設定します。



表 6: DPDK パラメータ

パラメータ	説明
<b>VPP_DPDK_BUFFERS=value</b>	<p>DPDK バッファの数を指定します。最小バッファは 32,000 で、最大値は VM サイズに基づきます。デフォルトのバッファ数は 128,000 です。</p> <p>DPDK バッファを 200,000 に設定するには、次の例を使用します。</p> <pre>VPP_DPDK_BUFFERS=200000</pre>
<b>VPP_DPDK_RX_QUEUES=value</b>	<p>特定の設定がないすべてのインターフェイスの RX キューの数を指定します。有効な値の範囲は、インターフェイスのタイプとホストの設定に応じて 1 ~ 64 になります。デフォルト値は VPP_CPU_WORKER_COUNT から計算され、各ワーカーに少なくとも 1 つのキューが割り当てられた状態でキューの全体的な数を最小限に抑えます。</p> <p>次の例を使用して、デフォルトの RX キューの数を 2 に設定します。</p> <p>(注) 一致しない値は、インターフェイスのタイプとホストの設定に基づいて設定できます。</p> <pre>VPP_DPDK_RX_QUEUES=2</pre>
<b>VPP_DPDK_TX_QUEUES=value</b>	<p>特定の設定がないすべてのインターフェイスの TX キューの数を指定します。有効な値の範囲は、インターフェイスのタイプとホストの設定に応じて 1 ~ 64 になります。デフォルト値は VPP_DPDK_RX_QUEUES です。</p> <p>次の例を使用して、デフォルトの TX キューの数を 4 に設定します。</p> <p>(注) 一致しない値は、インターフェイスのタイプとホストの設定に基づいて設定できます。</p> <pre>VPP_DPDK_TX_QUEUES=4</pre>

パラメータ	説明
<code>VPP_DPK_RX_DESCS=value</code>	<p>特定の設定がないすべてのインターフェイスの RX 記述子の数を指定します。有効な値の範囲は、インターフェイスのタイプとホストの設定に応じて 128 ~ 128,000 になります。デフォルト値は、指定されていないドライバ依存の値です。</p> <p>次の例を使用して、RX 記述子のデフォルトの数を 256 に設定します。</p> <p>(注) 一致しない値は、インターフェイスのタイプとホストの設定に基づいて設定できます。</p> <p><code>VPP_DPK_RX_DESCS=256</code></p>
<code>VPP_DPK_TX_DESCS=value</code>	<p>特定の設定がないすべてのインターフェイスの TX 記述子の数を指定します。有効な値の範囲は、インターフェイスのタイプとホストの設定に応じて 128 ~ 128,000 になります。デフォルト値は、<code>VPP_DPK_RX_DESCS</code> または指定されていないドライバ依存の値です。</p> <p>次の例を使用して、デフォルトの TX 記述子の数を 512 に設定します。</p> <p>(注) 一致しない値は、インターフェイスのタイプとホストの設定に基づいて設定できます。</p> <p><code>VPP_DPK_TX_DESCS=512</code></p>

### インターフェイス固有の設定

表に示した次のパラメータは、個々のインターフェイスを調整します。これらのパラメータは、デフォルトの DPDK 設定（該当する場合）もオーバーライドします。

表 7: インターフェイスのパラメータ

パラメータ	説明
<code>&lt;ROLE&gt;_VPP_RX_QUEUES=value</code>	<p>インターフェイスロールの RX キューの数を指定します。有効な値の範囲は、インターフェイスのタイプとホストの設定に応じて 1 ~ 64 になります。デフォルト値は未指定です。</p> <p>サービスポート 1 ~ 2 の RX キューの数を設定するには、次の例を使用します。</p> <p><code>SERVICE1_INTERFACE_VPP_RX_QUEUES=2</code></p>

パラメータ	説明
<code>&lt;ROLE&gt;_VPP_TX_QUEUES=value</code>	インターフェイスマジュールの TX キューの数を指定します。有効な値の範囲は、インターフェイスマジュールのタイプとホストの設定に応じて 1 ~ 64 になります。デフォルト値は <code>&lt;ROLE&gt;_VPP_RX_QUEUES</code> 、または <code>unspecified</code> です。
<code>&lt;ROLE&gt;_VPP_RX_DESCS=value</code>	インターフェイスマジュールの RX 記述子の数を指定します。有効な値の範囲は、インターフェイスマジュールのタイプとホストの設定に応じて 128 ~ 128,000 になります。デフォルト値は未指定です。  サービスポート 1 の RX 記述子の数を 1,024 に設定するには、次の例を使用します。  <code>SERVICE1_INTERFACE_VPP_RX_DESCS=1024</code>
<code>&lt;ROLE&gt;_VPP_TX_DESCS=value</code>	インターフェイスマジュールの RX 記述子の数を指定します。有効な値の範囲は、インターフェイスマジュールのタイプとホストの設定に応じて 128 ~ 128,000 になります。デフォルト値は <code>&lt;ROLE&gt;_VPP_RX_DESCS</code> 、または <code>unspecified</code> です。
<code>&lt;ROLE&gt;_VPP_WORKER_LIST=value</code>	インターフェイスマジュールのワーカースレッドを指定します。ワーカーリストは、Linux プロセッサ番号をカンマで区切ったリストです。有効な値は 0 ~ (CPU の数から 1 を差し引いた数) です。デフォルトは、すべてのソケットに割り当てられるラウンドロビン番号であり、すべてのソケットの最初のコアと最初のソケットの 2 番目のコアをスキップします。  サービスポート 1 のワーカースレッドリストを Linux プロセッサ番号 2 および 3 に設定するには、次の例を使用します。  <code>SERVICE1_INTERFACE_VPP_WORKER_LIST=2,3</code>

## ソフトウェアインストールおよびネットワーク展開

このガイドでは、VPC-SI が正しくインストールされ、市販 (COTS) サーバー上の仮想マシン (VM) 上で実行されていることを前提としています。

サポートされているオペレーティングシステムおよびハイパーバイザパッケージとプラットフォーム設定の詳細については、シスコの担当者にお問い合わせください。シスコのアドバンスドサービス（AS）グループは、VPC-SI 製品のコンサルティング、インストール、およびネットワーク展開のサービスを提供しています。



## 第 2 章

# システムの運用と設定

シスコ VPC-SI には、モバイルパケット コア ネットワーク用のサブスクリバ管理サービスが備わっています。

StarOS コマンドライン インターフェイス (CLI) に接続して設定プロセスを開始する前に、StarOS がこれらのサービスをどのようにサポートするかを理解しておく と 便利 です。この章では、StarOS を設定する前に考慮すべき用語と背景情報について説明します。

- [用語 \(29 ページ\)](#)
- [システムがコンテキストを選択する方法 \(33 ページ\)](#)
- [コンフィギュレーション ファイルの概要 \(36 ページ\)](#)
- [IP アドレスの表記法 \(37 ページ\)](#)
- [英数字の文字列 \(39 ページ\)](#)

## 用語

この項では、このガイド全体で使用されている重要な用語を定義します。

## コンテキスト

コンテキストは、さまざまな物理ポート、論理 IP インターフェイス、およびサービスに関連する設定パラメータの論理的なグループ化またはマッピングです。コンテキストは、バーチャルプライベート ネットワーク (VPN) と考えることができます。

システムには複数コンテキストを設定することができます。各コンテキストは、他のコンテキストとは独立して設定され、動作します。コンテキストが作成されると、管理ユーザーはそのコンテキストのサービス、論理 IP インターフェイス、およびサブスクリバを設定した後、論理インターフェイスを物理ポートにバインドできます。

また、コンテキストにドメインエイリアスを割り当てることもできます。サブスクリバのドメイン名が、コンテキストに設定されているエイリアス名のいずれかと一致する場合は、そのコンテキストが使用されます。

## 論理インターフェイス

ポートがユーザデータのフローを許可できるようにするには、論理インターフェイスと呼ばれる StarOS の仮想回路またはトンネルにポートを関連付ける必要があります。StarOS 内では、論理インターフェイスはレイヤ 3 IP アドレッシングなどの上位層のプロトコル転送を行う仮想ルータインターフェイスに関連付けられた名前付きのインターフェイスです。インターフェイスは、VPN コンテキストの一部として設定され、仮想インターフェイスをネットワークにブリッジするために使用される物理ポートから独立しています。

論理インターフェイスは、イーサネット + ppp + トンネルアドレスに関連付けられており、設定プロセス時に特定のポートにバインドされます。論理インターフェイスは、バインディングによりサービスに関連付けられます。サービスは、特定の論理インターフェイスに対して設定されている IP アドレスにバインドされます。関連付けられている場合、インターフェイスはサービスによって有効化された機能の特性を引き継ぎます。

簡易 IP データアプリケーションとモバイル IP データアプリケーションをサポートするように設定するための論理インターフェイスにはいくつかのタイプがあります。次に、これらの簡単な定義を示します。

## 管理インターフェイス

このインターフェイスは、管理ネットワークへの接続ポイントを提供します。このインターフェイスは、StarOS のコマンドラインインターフェイス (CLI) へのリモートアクセスをサポートしています。また、Simple Network Management Protocol (SNMP) を介したイベント通知にも対応しています。

## バインディング

バインドは、システム内の要素間の関連付けです。バインドには、動的と静的という2つのタイプがあります。

静的バインドは、システム設定によって実現されます。静的バインドは、次のように関連付けを行います。

- 物理ポートに対して（特定のコンテキスト内で設定された）特定の論理インターフェイス。インターフェイスがバインドされると、トラフィックは物理的に定義された回路であるかのように、コンテキストを経由することができます。静的バインドは、任意のインターフェイスおよびポートタイプでのカプセル化方式をサポートします。
- 同じコンテキスト内の論理インターフェイスに割り当てられた IP アドレスへのサービス。これにより、インターフェイスは、サービスに必要な特性（つまり、プロトコルをサポートする）を引き継ぐことができます。

動的バインドは、プロファイルまたはシステムパラメータの設定に基づいて、サブスライバを特定の出力コンテキストに関連付けます。これにより、ワイヤレスキャリアが複数のサービスをサポートし、複数のネットワークへのシームレスな接続を容易にすることができるため、導入の柔軟性が高まります。

管理ポートは、ローカルコンテキストでのみバインドできます。トラフィックポートまたはサブスクライバポートは、非ローカルコンテキストでのみバインドできます。

## サービス

特定の機能を有効にするには、コンテキスト内でサービスを設定します。次に、システムで設定できるサービスの例を示します。これは、ライセンスの可用性とプラットフォームのタイプによって異なります。

- ゲートウェイ GPRS サポートノード (GGSN) サービス
- サービング GPRS サポートノード (SGSN) サービス
- パケット データ サービング ノード (PDSN) サービス
- ホームエージェント (HA) サービス
- レイヤ 2 トンネリング プロトコル アクセス コンセントレータ (LAC) サービス
- ダイナミックホスト制御プロトコル (DHCP) サービス
- モビリティ マネージメント エンティティ (MME) サービス
- PDN ゲートウェイ (P-GW) サービス
- サービングゲートウェイ (S-GW) サービス
- インテリジェントポリシー制御機能 (IPCF) サービス (PCC-Service、PCC-Policy、PCC-AF)

## AAA サーバー

認証、許可、およびアカウントリング (AAA) サーバーは、プロファイルを保存し、認証を実行し、各モバイルデータサブスクライバのアカウントリングレコードを維持します。AAA サーバーは、AAA インターフェイスを介してシステムと通信します。システムでは、AAA サーバーへの最大 128 のインターフェイス設定がサポートされています。

モバイル IP の場合は、外部 AAA (FAAA) サーバーとホーム AAA (HAAA) サーバーが存在する可能性があることに注意することが重要です。通常、FAAA サーバーはキャリアのネットワークに存在します。HAAA サーバーは、キャリアかホームネットワークのいずれかによって所有および制御されていることがあります。HAAA サーバーがホームネットワークによって所有および制御されている場合、アカウントリングデータは AAA プロキシサーバー経由でキャリアに転送されます。



---

**重要** モバイル IP のサポートは、ホームエージェント (HA) を含むライセンスバンドルの可用性と購入によって異なります。

---

## サブスクリバ

サブスクリバは、サービスのエンドユーザーです。システムを介してインターネット、ホームネットワーク、またはパブリックネットワークへのアクセスを取得します。

サブスクリバには、主に次の3つのタイプがあります。

- **RADIUS ベースのサブスクリバ**：最も一般的なタイプのサブスクリバであるこれらのユーザーは、International Mobile Subscriber Identity (IMSI) 番号、電子シリアル番号 (ESN)、またはドメイン名やユーザー名によって識別されます。これらは、RADIUS AAA サーバーで設定され、認証されます。

認証が成功すると、サブスクリバプロファイルに含まれているさまざまな属性が返されます。属性は、セッションパラメータの設定（たとえば、プロトコル設定と IP アドレスの割り当て方法）、およびサブスクリバが持っている権限などを決定します。



---

**重要** RADIUS AAA サーバーからシステムが受信した属性設定は、システムに設定されているローカルサブスクリバの属性とパラメータよりも優先されます。

---

- **ローカルサブスクリバ**：これらは、主にテスト目的で使用されるサブスクリバであり、特定のコンテキスト内で設定および認証されます。RADIUS ベースのサブスクリバとは異なり、ローカルサブスクリバのユーザープロファイル (RADIUS ベースのサブスクリバによって使用される属性を含む) は、それらが作成されたコンテキスト内で設定されます。

ローカルサブスクリバのプロファイルが最初に作成されると、そのサブスクリバの属性はシステムのデフォルトに設定されます。すべてのサブスクリバのプロファイルに同じデフォルト設定が適用されます。これには、各システムコンテキストのシステムによって自動的に作成される *default* という名前のサブスクリバが含まれます。ローカルプロファイルの属性を設定すると、サブスクリバごとに変更が行われます。



---

**重要** ローカルサブスクリバ用に設定された属性は、コンテキストレベルのパラメータよりも優先されます。ただし、RADIUS AAA サーバーから返された属性によって、これらの属性が上書きされる可能性があります。

---

- **管理サブスクリバ**：管理ユーザーは、CLI を通じてシステムをモニター、制御、および設定できる権限を持つユーザーです。管理は、システムコンソールポートを介してローカルで実行されるか、Telnet またはセキュアシェル (SSH) プロトコルを使用してリモートで実行されます。管理ユーザーは通常、ローカルコンテキスト内でローカルサブスクリバとして設定されます。これは、システム管理と管理専用で使用されます。ローカルサブスクリバと同様に、管理者サブスクリバのユーザープロファイルは、サブスクリバが作成されたコンテキスト内（この場合はローカルコンテキスト内）で設定されます。た



だし、ローカルコンテキスト内または TACACS+ 内に AAA 設定が存在する場合、管理サブスクリバは RADIUS を介してリモートで認証されることもあります。

## システムがコンテキストを選択する方法

この項では、コンテキストレベルの管理ユーザーまたはサブスクリバのセッションに使用するコンテキストを決定するプロセスについて説明します。このプロセスを理解することで、設定が必要なコンテキストとインターフェイスの数の設定をより適切に計画できるようになります。

### コンテキストレベルの管理ユーザーセッションのコンテキスト選択

システムは、管理目的で具体的に使用する *local* という名前のコンテキストで設定されます。コンテキストレベルの管理ユーザー（コンテキスト内で設定されている）のコンテキスト選択プロセスは、MIOの管理ポートがローカルコンテキストだけに関連付けられているため、簡素化されています。したがって、システム全体の管理を担当するコンテキストレベルの管理ユーザーの送信元と接続先のコンテキストは、常にローカルコンテキストである必要があります。

コンテキストレベルの管理ユーザーは、非ローカルのコンテキストで作成できます。これらの管理アカウントには、作成されたコンテキストでのみ権限が付与されます。このタイプの管理アカウントは、そのコンテキストでローカル接続が有効になっている場合（SSHD など）、属しているコンテキストのポートに直接接続できます。

すべての FTP 接続または SFTP 接続については、MIO 管理インターフェイスを介して接続する必要があります。非ローカルのコンテキストアカウントとして SFTP または FTP を利用する場合は、*username@contextname* のユーザー名のシンタックスを使用する必要があります。



**重要** FTP はサポートされていません。

ローカル認証を提供するようにシステムを設定している場合、またはコンテキストレベルの管理ユーザーを認証するために AAA サーバーと連携する場合に、コンテキスト選択プロセスはより複雑になります。

システムは、コンテキストレベルの管理ユーザーをローカルに設定する（つまり、プロファイルが設定され、独自のメモリに保存される）、またはリモートで AAA サーバー上に設定するという柔軟性を提供します。ローカルに設定されたユーザーがシステムにログオンしようとすると、システムは認証を実行します。AAA サーバーでユーザープロファイルを設定している場合、システムは AAA サーバーに接続して認証を実行する方法を決定する必要があります。これは、セッションの AAA コンテキストを決定することによって行われます。

次の表とフローチャートは、コンテキストレベルの管理ユーザーに対して AAA コンテキストを選択するためにシステムが使用するプロセスを示しています。表の項目は、フローチャート内の丸で囲まれた番号に対応しています。

図 1: コンテキストレベルの管理ユーザーの AAA コンテキスト

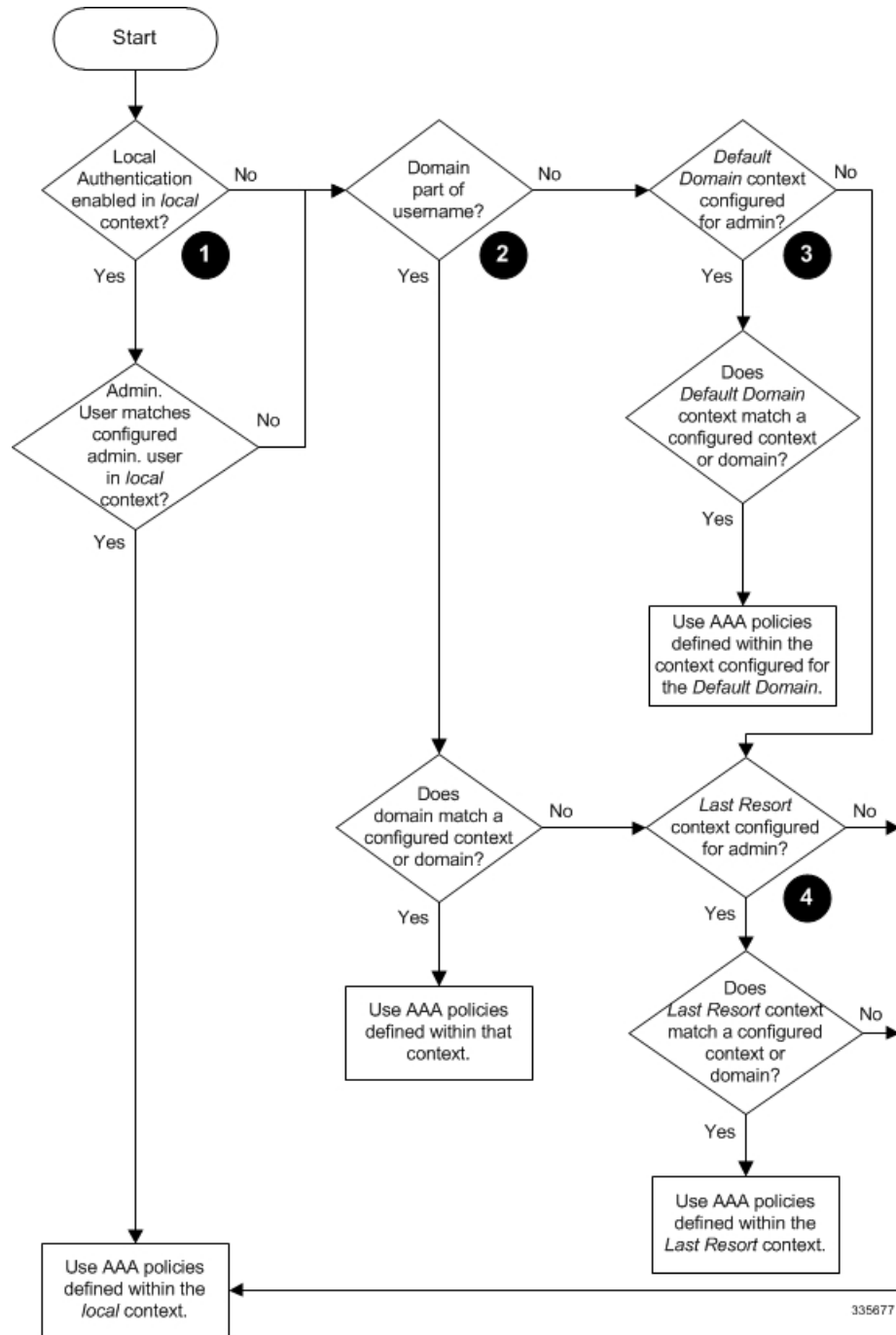


表 8: コンテキストレベルの管理ユーザーの AAA コンテキストにおける選択

項目	説明
1	<p>認証時に、ローカルコンテキストでローカル認証が有効になっているかどうかシステムによって判別されます。</p> <p>有効になっている場合、システムはローカルコンテキストで管理ユーザーの認証を試行します。有効になっていない場合、この表の項目 2 に進みます。</p> <p>管理ユーザーのユーザー名が設定されている場合、ローカルコンテキスト内の AAA 設定を使用して認証が実行されます。設定されていない場合、この表の項目 2 に進みます。</p>
2	<p>システムでローカル認証が無効になっている場合、または管理ユーザーのユーザー名がローカルコンテキストで設定されていない場合、システムはユーザー名の一部としてドメインが受信されたかどうかを判断します。</p> <p>ドメインが存在し、設定されているコンテキストまたはドメインの名前に一致する場合、システムはそのコンテキスト内で AAA 設定を使用します。</p> <p>ドメインが存在し、設定されているコンテキストまたはドメインの名前と一致しない場合、この表の項目 4 に進みます。</p> <p>ユーザー名の一部としてドメインが存在しない場合、この表の項目 3 に進みます。</p>
3	<p>ユーザー名にドメインが指定されていない場合、またはドメインが認識されていない場合、システムは AAA 管理者のデフォルトドメインが設定されているかどうかを判別します。</p> <p>デフォルトドメインが設定されており、設定済みコンテキストと一致する場合は、AAA 管理者のデフォルトドメインにおけるコンテキスト内の AAA 設定が使用されます。</p> <p>デフォルトドメインが設定されていないか、設定済みコンテキストまたはドメインと一致しない場合は、次の項目 4 に進みます。</p>
4	<p>ドメインがユーザー名の一部として指定されていても、設定済みコンテキストと一致しなかった場合、またはドメインがユーザー名の一部として指定されていない場合、システムは AAA 管理者のラストリゾート コンテキスト パラメータが設定されているかどうかを判断します。</p> <p>ラストリゾートコンテキストが設定されており、設定済みコンテキストと一致する場合、そのコンテキスト内の AAA 設定が使用されます。</p> <p>ラストリゾートコンテキストが設定されていない場合、または設定されたコンテキストやドメインと一致しない場合、ローカルコンテキスト内の AAA 設定が使用されます。</p>

リリース 21.4 以降（信頼できるビルドのみ）：

- ユーザーは、それぞれのコンテキストインターフェイスを使用してのみシステムにアクセスできます。
- ユーザーが別のコンテキストインターフェイスを使用してそれぞれのコンテキストにログインしようとする、そのユーザーは拒否されます。

- ユーザーが「authorized-keys」または「allowusers」のいずれかのコンテキストに設定されているかどうかに関係なく、この機能では、独自のコンテキストインターフェイス以外のコンテキストインターフェイスを介してログインを試行すると、これらのユーザーは拒否されます。
- ローカル以外のコンテキストに設定されているユーザーは、ログインを試行するコンテキストを指定する必要があります。次に例を示します。

```
ssh username@ctx_name@ctx_ip_addr
```

## サブスクライバセッションのコンテキスト選択

管理ユーザーの場合よりも、サブスクライバセッションのコンテキスト選択プロセスとの関係があります。特定製品のサブスクライバセッション コンテキストの選択情報は、個々の製品の「*Administration Guide*」に記載されています。

## コンフィギュレーション ファイルの概要

システムでは、設定可能なパラメータを変更するためのファイルまたはスクリプトの使用がサポートされています。ファイルをオフラインのシステム設定に使用すると、複数のシステムでパラメータを設定するのにかかる時間が短縮されます。

システム コンフィギュレーション ファイルは、コマンドと設定パラメータを含む ASCII テキストファイルです。コンフィギュレーション ファイルを適用すると、システムはファイルを行単位で解析し、シンタックスをテストしてコマンドを実行します。シンタックスが正しくない場合は、CLI にメッセージが表示され、システムは次のコマンドに進みます。# で始まる行は注釈と見なされ、無視されます。



**重要** `grep` キーワードおよび `more` キーワードとともに使用されるパイプ (`|`) は、コンフィギュレーションファイルの処理においてエラーを引き起こす可能性があります。したがって、システムは処理中にパイプのあるキーワードを自動的に無視します。



**重要** コンフィギュレーションファイルは常にUNIX形式で保存してください。そうしないと、コンフィギュレーションファイルの処理を妨げるエラーが発生する可能性があります。

ファイル内のコマンドと設定データは、CLI プロンプトで入力した場合と同様に整理され、フォーマットされます。たとえば、CLI で `source` というコンテキストを作成する場合は、それぞれのプロンプトで次のコマンドを入力します。

```
[local]host_name# config
[local]host_name(config)# context source
[source]host_name(config-ctx)# end
```

コンフィギュレーションファイルを使用して *source* というコンテキストを作成するには、テキストエディタを使用して、次のもので構成される新しいファイルを作成します。

```
config
  context source
end
```

コンフィギュレーションファイルを使用する際に考慮すべき重要な点がいくつかあります。

- システムは、起動プロセスの最後にコンフィギュレーションファイルを自動的に適用します。システムを初めて起動した後、作成したコンフィギュレーションファイルと、ネットワークのニーズに合わせて調整したコンフィギュレーションファイルを適用できます。システムがコンフィギュレーションファイルを使用するには、『*Software Management Operations*』に記載されている手順に従ってシステムの起動パラメータを変更します。
- 起動プロセス中に適用されるだけでなく、CLIプロンプトで適切なコマンドを実行することで、いつでもコンフィギュレーションファイルを手動で適用できます。『*Software Management Operations*』の手順を参照してください。



---

**重要** 起動プロセス後にコンフィギュレーションファイルが適用しても、ブートプロセスの一部としてロードされたコンフィギュレーションは削除されません。重複しているコマンドだけが上書きされます。

---

- コンフィギュレーションファイルは、次の場所のいずれかに保存することができます。
  - **USB メモリスティック** : アクティブな MIO (/usb1) 上の USB ポートを介してサポートされる。
  - **ネットワークサーバー** : システムがセキュアファイル転送プロトコル (SFTP) を使用してアクセスできるネットワーク上のワークステーションまたはサーバー。これは、複数のシステムが同じ設定を必要とする大規模なネットワークの展開に推奨される。
  - **/flash** : ストレージが制限されているソリッドステートデバイス。
  - **raid** : 内部 RAID ストレージ。
- CLIセッション中に行った設定変更を保存するたびに、コンフィギュレーションファイルとして使用可能なファイルにそれらの設定を保存できます。

## IP アドレスの表記法

CLIを介してポートインターフェイスを設定する場合は、IPアドレスを入力する必要があります。CLIは常にIPv4アドレスを受け入れ、場合によってはIPv6アドレスを代替として受け入れます。

一部の設定コマンドでは、CLIはCIDR表記も受け入れます。受け入れ可能なIPアドレスの表記法の形式を確認するには、常にCLIコマンドのオンラインヘルプを参照してください。

## IPv4 ドット付き 10 進表記

インターネットプロトコルバージョン4 (IPv4) アドレスは、4つのオクテットに分割された32ビットで構成されます。これらの4つのオクテットは、0～255の範囲の10進数の番号で記述され、各番号の間に終止符のデリミタ（ドット）を含む文字列として連結されます。

たとえば、通常はホスト名 localhost が割り当てられているループバック インターフェイスのアドレスは127.0.0.1です。これは、4つのバイナリオクテット01111111、00000000、00000000、および00000001で構成され、完全な32ビットアドレスを形成します。

IPv4では、インターネットプロトコルアドレスに32ビットを使用できるため、 $2^{32}$  (4,294,967,296) アドレスがサポートされます。

## IPv6 コロン区切り 16 進表記

インターネットプロトコルバージョン6 (IPv6) アドレスには、64ビットのネットワークおよび64ビットのホストアドレスといった2つの論理的な部分があります。IPv6アドレスは、16ビット16進数の値のコロン (:) で区切られた8つのグループで表されます

完全なIPv6アドレスの一般的な例は、2001:0db8:85a3:0000:0000:8a2e:0370:7334です。

16進数は大文字と小文字が区別されません。

128ビットのIPv6アドレスは、次のルールを使用して省略できます。

- 16ビット値の先頭のゼロは省略できます。たとえば、アドレス fe80:0000:0000:0000:0202:b3ff:fe1e:8329 は、fe80:0:0:0:202:b3ff:fe1e:8329 として記述されます。
- アドレス内の連続したゼロの1つのグループは、二重コロンで置き換えることができます。たとえば、fe80:0:0:0:202:b3ff:fe1e:8329 は、fe80::202:b3ff:fe1e:8329 になります。

IPv6では、インターネットプロトコルアドレスに128ビットを使用でき、 $2^{128}$  ビット (340,282,366,920,938,000,000,000,000,000,000,000,000,000) のインターネットアドレスをサポートできます。

## CIDR 表記

Classless Inter-Domain Routing (CIDR) の表記法は、インターネットプロトコルアドレスとそれに関連付けられているルーティングプレフィックスのコンパクトな仕様です。これは、ネットワークアーキテクチャでIPv4とIPv6の両方のアドレッシングに使用されます。

CIDRは、IPアドレスの解釈のためのビット単位のプレフィックススペースの標準規格です。アドレスのブロックを単一のルーティングテーブルエントリにグループ化することにより、ルー

ティングを容易にします。これらのグループ（CIDR ブロック）は、IP アドレスのバイナリ表記で最初のビットシーケンスを共有します。

CIDR 表記は、IP アドレスとプレフィックスのサイズから構築され、後者はルーティングプレフィックスの先頭の 1 ビットの数になります。IP アドレスは、IPv4 または IPv6 の標準規格に従って表されます。これには、区切り文字、スラッシュ (/) 文字、および 10 進数で表されるプレフィックスサイズが続きます。

アドレスは、単一、個別、インターフェイスアドレス、またはネットワーク全体の開始アドレスを表すことができます。後者の場合、CIDR 表記は、ネットワークのアドレスブロック割り当てを指定します。ネットワークの最大サイズは、プレフィックス以下の残りの最下位ビットによって有効になるアドレスの数で表されます。これは、ホスト識別子と呼ばれることがよくあります。

次に例を示します。

- アドレス指定 209.165.200.224/27 は、特定の IPv4 アドレスとそれに関連付けられたルーティングプレフィックス 209.165.200.0、または同等のサブネットマスク 255.255.255.224 を表します。
- IPv4 ブロック 192.168.0.0/22 は、192.168.0.0 から 192.168.3.255 までの 1024 IPv4 アドレスを表します。
- IPv6 ブロック 2001:DB8::/48 は、2001:DB8:0:0:0:0:0:0 から 2001:DB8:0:FFFF:FFFF:FFFF:FFFF:FFFF までの IPv6 アドレスを表します。
- ::1/128 は IPv6 ループバックアドレスを表します。プレフィックスサイズは 128 です。これはアドレス自体のサイズであり、このファシリティが 1 つのアドレスだけで構成されていることを示しています。

マスクまたはプレフィックスによって定義されたサブネットのアドレスの数は、 $2^{\text{address size} - \text{mask}}$  として計算することができます。この場合、IPv4 のアドレスサイズは 32、IPv6 のアドレスサイズは 128 です。たとえば、IPv4 では、/29 のマスクは  $2^{32-29} = 2^3 = 8$  addresses を与えます。

## 英数字の文字列

一部の CLI コマンドでは、値を定義するために英数字文字列を入力する必要があります。この文字列は、定義された最小長および最大長（文字数）を持つ連続した英数字のコレクションです。

## 文字セット

英数字セットは、アルファベット（ラテン文字）や数字（アラビア数字）の組み合わせです。このセットは、0～9 の数字、A～Z（大文字）、a～z（小文字）で構成されます。アンダースコア文字（\_）とダッシュ/ハイフン（-）は、英数字の文字セットの一部と見なされます。

空白文字（空白文字またはスペース文字）は、通常、時間や日付スタンプなどの特定のルール定義形式を除き、英数字文字列では回避する必要があります。

以下に示す通り、英数字文字列には次の「特殊」文字を使用しないてください。

- & (アンパサンド)
- ' (アポストロフィ)
- <> (矢印ブラケット) [以下の例外を参照してください]
- \* (アスタリスク) [以下のワイルドカードの例外を参照してください]
- { } (波カッコ)
- [ ] (角カッコ)
- \$ (ドル記号) [以下のワイルドカード例外を参照してください]
- ! (感嘆符) [以下の例外を参照してください]
- ( ) (丸カッコ)
- % (パーセント) [以下の例外を参照してください]
- # (ポンド記号) [以下の例外を参照してください]
- ? (疑問符)
- ' (引用符：シングル)
- " (引用符：ダブル)
- ; (セミコロン)
- \ (バックスラッシュ) [以下の例外を参照してください]
- / (普通のスラッシュ) [以下の例外を参照してください]
- ~ (チルダ)
- | (縦棒) [以下の例外を参照してください]

ruledef、APN、ライセンスキー、およびその他の設定/表示パラメータで入力された文字列には、次の文字が表示される場合があります。

- <> (矢印ブラケット) [より小さい or より大きい]
- \* (アスタリスク) [ワイルドカード]
- : (コロン)
- \$ (ドル記号) [ワイルドカード]
- 。 (ドット)
- = (等号)
- ! (感嘆符)



- % (パーセント)
- / (普通のスラッシュ)
- | (縦棒)

次の文字を使用して、グローバル AAA 機能のユーザー名からドメインを区切ることができます。

- @ (アットマーク)
- (ダッシュまたはハイフン)
- # (ハッシュまたはポンド記号)
- % [パーセント]
- \ (バックスラッシュ) [二重スラッシュ 「\\」 として入力する必要があります]
- / (普通のスラッシュ)

## 引用符付き文字列

説明文で単語の間にスペースを使用する必要がある場合は、二重引用符 ("") 内に文字列を入力する必要があります。次に例を示します。

```
interface "Rack 3 Chassis 1 port 5/2"
```





## 第 3 章

### 使用する前に

- [StarOS CLI を使用した初期設定 \(43 ページ\)](#)
- [システム管理ユーザーの設定 \(45 ページ\)](#)
- [リモートアクセス用のシステムの設定 \(47 ページ\)](#)
- [SSH オプションの設定 \(49 ページ\)](#)
- [2 番目の IP アドレスを使用した管理インターフェイスの設定 \(61 ページ\)](#)
- [Open SSH から Cisco SSH へのアップグレードと移行 \(62 ページ\)](#)
- [VM ハードウェアの検証 \(65 ページ\)](#)

## StarOS CLI を使用した初期設定

初期設定は次のように構成されています。

- コンテキストレベルのセキュリティ管理者とホスト名の設定
- vNIC でのイーサネット インターフェイスの設定
- Telnet、SSH、または FTP（セキュアまたは非セキュア）によるリモート CLI アクセスのためのシステム設定

この項では、CLI を使用してこれらのタスクを実行するための手順を説明します。

**ステップ 1** ハイパーバイザを介してコンソールポートにログインしてください。

**ステップ 2** CLI プロンプトで、次のように入力します。

```
[local]host_name configure[local]host_name(config)
```

**ステップ 3** 次のコマンドを入力してコンテキスト コンフィギュレーション モードを開始します。

```
[local]host_name(config) context local[local]host_name(config-ctx)
```

ローカルコンテキストは、システムの管理コンテキストです。コンテキストを使用すると、サービスまたはインターフェイスを論理的にグループ化することができます。1つのコンテキストは複数のサービスで構成でき、複数のインターフェイスにバインドできます。

**ステップ4** システムのコンテキストレベルのセキュリティ管理者を設定するには、次のコマンドを入力します。

```
administrator user_name [ encrypted ] password password | [ ecs ] [ expiry-date
date_time ] [ ftp ] [ li-administration ] [ nocli ] [ noecs ]      ]
```

(注) 初期設定時にコンテキストレベルのセキュリティ管理者を設定する必要があります。初期設定プロセスが完了し、CLIセッションを終了した後、セキュリティ管理者が設定されていない場合は、CLIアクセスがロックされます。このコマンドの詳細については、『*Command Line Interface Reference*』の「*Context Configuration Mode Commands*」の章を参照してください。

**ステップ5** プロンプトで次のコマンドを入力して、コンテキストのコンフィギュレーションモードを終了します。

```
[local]host_name(config-ctx) exit
[local]host_name(config)
```

**ステップ6** 次のコマンドを入力して、システムがネットワーク上で認識されるホスト名を設定します。

```
[local]host_name(config) system hostname host_name
```

*host\_name* は、ネットワーク上でシステムが認識される名前です。ホスト名は、大文字と小文字が区別される 1～63 文字の英数字文字列です。デフォルトのホスト名は「*qvpc-si*」です。

**ステップ7** vNIC 上のネットワーク インターフェイスを次のように設定します。

a) 次のコマンドを入力して、コンテキスト コンフィギュレーション モードを開始します。

```
[local]host_name(config) context local
[local]host_name(config-ctx)
```

b) インターフェイスの名前を指定するには、次のコマンドを入力します。

```
[local]host_name(config-ctx) interface interface_name
```

*interface\_name* は、大文字と小文字が区別される 1～79 文字の英数字の文字列で表されるインターフェイスの名前です。システムがイーサネット インターフェイスのコンフィギュレーションモードを開始する、次のプロンプトが表示されます。

```
[local]host_name(config-if-eth)
```

c) 次のコマンドを入力して、前のステップで設定したインターフェイスの IP アドレスを設定します。

```
{ ip address | ipv6 address } ipaddress subnetmask
```

(注) クイック セットアップ ウィザードで誤って設定されたアドレスまたはサブネットを修正するためにこのコマンドを実行する場合は、デフォルトルートとポートバインドの設定を確認する必要があります。この手順のステップ 11 とステップ 6 を使用します。問題がある場合は、ステップ 7e～7k を実行して情報を再設定します。

d) 次のコマンドを入力して、イーサネット インターフェイスのコンフィギュレーションモードを終了します。

```
[local]host_name(config-if-eth) exit
[local]host_name(config-ctx)
```

e) 必要に応じてスタティックルートを設定して、システムをデフォルトゲートウェイに指定します。次のコマンドを入力します。

```
{ ip | ipv6 } route gw_address interface_name
```

- f) コンテキストのコンフィギュレーションモードを終了するには、次のように入力します。

```
[local]host_name(config-ctx) exit  
[local]host_name(config)
```

- g) イーサネットポートのコンフィギュレーションモードを開始します。

```
port ethernet slot/port
```

VPC の場合、スロット番号は常に「1」です。vNIC トラフィックポートは 10 ～ 21 です。ポート 1 は管理ポートです。

- h) ステップ 7b で作成したインターフェイスにポートをバインドします。バインドにより、ポートとそのすべての設定がインターフェイスに関連付けられます。次のコマンドを入力します。

```
[local]host_name(config-port-slot/port) bind interface interface_name local  
[local]host_name(config-port-slot/port) no shutdown
```

*interface\_name* は、ステップ 7b で設定したインターフェイスの名前です。

- i) 次のコマンドを入力して、イーサネットインターフェイスのコンフィギュレーションモードを終了します。

```
[local]host_name(config-port-slot/port) exit  
[local]host_name(config)
```

(注) 管理ポートは、VLAN もサポートしています。詳細については、「インターフェイスとポート」の章の「VLAN」の項を参照してください。

2 番目の IP アドレスを使用して vNIC 管理インターフェイスを設定する手順については、以下を参照してください。

---

## システム管理ユーザーの設定

この項では、セキュリティ管理者がユーザーアカウントを制御できるようにするセキュリティ機能の一部について説明します。

### 同時 CLI セッション数の制限

セキュリティ管理者は同時対話型 CLI セッションの数を制限できます。同時対話型セッションの数を制限すると、システム全体のリソースの消費量が削減されます。また、ユーザーがすでに使用されている機密ユーザー情報にアクセスする可能性を防ぎます。

ほとんどの特権アカウントでは、複数の同時ログインは必要ありません。



- 
- (注) 21.9 以降のリリースでは、1 つの CLI セッションでの複数のチャンネルはサポートされていません。
-



---

**重要** すべての特権アカウントには、セッションの最大数を設定することを推奨します。

---

セキュリティ管理者は、その特定のユーザーアカウントに使用される認証方式に応じて、3つの異なる方法で同時インタラクティブ CLI セッションの数を制限できます。

StarOS は次の3つのログイン認証方式をサポートしています。

- TACACS+ サーバーユーザー
- ローカルユーザーのユーザー
- AAA コンテキストユーザー

TACACS+ サーバーユーザーの最大セッション数の設定の詳細については「[動作](#)」を参照してください。ローカルユーザーのユーザーと AAA コンテキストユーザーの最大セッション数の設定の詳細については「[コンテキストレベルの管理ユーザーの設定](#)」を参照してください。

各認証方式は、3つの認証方式のそれぞれが同じユーザー名を使用できるため、個別に設定する必要があります。

## CLI セッションの自動ログアウト

セキュリティ管理者は、特定のユーザーアカウントの自動ログアウトを設定できます。対話型 CLI セッションが使用可能な時間を分単位で制限すると、システム全体のリソースの消費量が削減されます。また、アイドル状態のままになっている端末ウィンドウで、ユーザーがユーザーアカウントにアクセスする可能性を防ぐこともできます。この項で説明されているすべての認証方式は、アイドルセッションタイムアウトの手法と絶対セッションタイムアウトの手法の両方をサポートしています。

ほとんどの特権アカウントは、無期限のログインタイムアウトの制限を必要としません。



---

**重要** すべての特権アカウントには、セッションタイムアウトを設定することを推奨します。

---

**show tacacs summary** コマンドと **show tacacs session id** コマンドのアイドルタイムアウトおよびセッションタイムアウトのフィールドを使用すると、管理者は特定のアカウントの自動ログアウトを設定できます。

**セッションタイムアウト**：セキュリティ管理者は、セッションが自動的に切断される前に、ユーザーがセッションにログオンできる最大時間を分単位で指定できます。

**アイドルタイムアウト**：セキュリティ管理者は、セッションが自動的に切断される前に、セッションがアイドル状態を維持できる最大時間を分単位で指定できます。



---

**重要** セッションタイムアウトとアイドルタイムアウトのフィールドは排他的ではありません。両方が指定されている場合は、低いセッションタイムアウトが常に最初に到達するため、アイドルタイムアウトは常にセッションタイムアウトよりも低くする必要があります。

---

対話型CLIセッションを使用できる最大時間を分単位で設定する方法の詳細については、『*CLI Reference*』の **dle-sessions threshold** コマンドと **clear tacacs sessions** CLI コマンド、および『*Statistics and Counter Reference*』の **show tacacs summary** と **show tacacs session id** を参照してください。

## リモートアクセス用のシステムの設定

リモートアクセス用にシステムを設定します。管理ユーザーは、管理ネットワークを介してリモートの場所からインスタンスにアクセスできます。

- Telnet
- セキュア シェル (SSH)
- File Transfer Protocol (FTP) (セキュアまたは非セキュア)
- Trivial File Transfer Protocol (TFTP)



(注) 2つの同時 telnet セッションがあり、1人の管理者が他の管理者がログに記録するコンテキストを削除した場合は、削除されたコンテキストの管理者が自動的にローカルコンテキストに退出させられることはありません。削除されたコンテキストはCLIプロンプトに引き続き表示されますが、コンテキスト固有のコマンドによってエラーが生成されます。



(注) セキュリティを最大限にするには、SSH v2 を使用します。



(注) FTP および telnet はサポートされていません。

**ステップ1** 次のコマンドを入力してコンテキスト コンフィギュレーション モードを開始します。

```
[local]cf_host_name(config) context local  
[local]cf_host_name(config-ctx)
```

**ステップ2** 必要に応じて、Telnet アクセスを許可するようにシステムを設定します。

```
[local]cf_host_name(config-ctx) server telnetd
```

**ステップ3** 必要に応じて、SSH アクセスを許可するようにシステムを設定します。

```
[local]cf_host_name(config-ctx) ssh generate key [ type v2-rsa ]
```

(注) **v2-rsa**は推奨されるキータイプです。

(注) **v2-dsa** キーワードは コンテキスト コンフィギュレーション モードの **ssh generate** CLI コマンド内に隠されています。以前のリリースでサポートされていたキーワードが後続のリリースでは隠されている可能性があります。システムは、以前のリリースで作成された既存のスクリプトや設定ファイル内の隠されたキーワードを引き続き解析します。ただし、新しいスクリプトやコンフィギュレーション ファイルで使用するために、コマンドシンタックスに隠されたキーワードは表示されなくなりました。疑問符 (?) を入力しても、ヘルプテキストの一部として隠しキーワードは表示されません。削除されたキーワードを指定すると、解析時にエラーメッセージが生成されます。

```
[local]cf_host_name(config-ctx) server sshd
[local]cf_host_name(config-sshd) subsystem sftp
[local]cf_host_name(config-sshd) exit
```

**ステップ 4** 必要に応じて、次のコマンドを入力して、FTP アクセスを許可するようにシステムを設定します。

```
[local]cf_host_name(config-ctx) server ftpd
```

**ステップ 5** 次のコマンドを入力して、コンフィギュレーション モードを終了します。

```
[local]cf_host_name(config-ctx) end
[local]cf_host_name
```

**ステップ 6** 次のコマンドを入力して、設定を確認します。

```
[local]cf_host_name show configuration
```

CLI 出力は、次の出力例のようになります。

```
context local
  interface interface_name
    ip address ipaddress subnetmask
    exit
  subscriber default
    exit
  administrator admin_name password admin_password
  server telnetd
  server ftpd
  ssh generate key
  server sshd
  subsystem sftp
  exit
port ethernet 1/1
  bind interface interface_name local
  exit
port ethernet 1/1
  no shutdown
  exit
snmp engine-id local 800007e580ed826c191ded2d3d
end
```

**ステップ 7** 次のコマンドを入力して、IP ルートの設定を確認します。

```
[local]cf_host_name show ip route
```

CLI 出力は、次の出力例のようになります。

```
"" indicates the Best or Used route.
Destination          Nexthop          Protocol  Prec Cost Interface
```



```
*0.0.0.0/0          ipaddress      static      1    0    vnic1
*network           0.0.0.0        connected  0    0    vnic1
```

**ステップ 8** 次のコマンドを入力して、インターフェイス バインディングを確認します。

```
[local]cf_host_name show ip interface name interface_name
```

*interface\_name* は、手順 7b で設定したインターフェイスの名前です。CLI 出力は、次の出力例のようになります。

```
Intf Name:          vnic1

Description:
IP State:           UP (Bound to 1/1 untagged, ifIndex 83951617)
IP Address:         ipaddress          Subnet Mask:      subnetmask
Bcast Address:     bcastaddress       MTU:              1500
Resoln Type:       ARP                 ARP timeout:      3600 secsL3 monitor LC-port
switchover:        DiasabledNumber of Secondary Addresses:  0
```

**ステップ 9** 「設定の確認と保存」の章の説明に従って、設定を保存します。

## SSH オプションの設定

### SSH ホストキー

SSH キーベースの認証では、誰に対しても表示が許可されている「公開」キーと、所有者のみが表示を許可されている別の「秘密」キーの、2つのキーを使用します。キーペアを作成し、ログインするデバイスに秘密キーを安全に保存して、ログインするシステム（VPC-SI）に公開キーを保存します。

SSH ホストキーは、指定された StarOS コンテキスト内で生成されます。コンテキストは、ユーザーインターフェイスに関連付けられています。

コンテキストに関連付けられている `sshd` サーバーにアクセスするための承認されたキーを持つ管理ユーザー名を設定または削除します。

### SSH キーのサイズ設定

グローバル コンフィギュレーション モードの `ssh key-size` CLI コマンドは、すべてのコンテキストの SSH キー生成のキーサイズを設定します（RSA ホストキーのみ）。

**ステップ 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。

```
[local]host_name# configure
[local]host_name(config)#
```

**ステップ 2** SSH キーのビットサイズを指定します。

```
[local]host_name(config)# ssh key-size { 2048 | 3072 | 4096 | 5120 | 6144 | 7168 | 9216 }
```

SSH キーのデフォルトのビットサイズは 2048 ビットです。

## SSH キー生成の待機時間の設定

SSH キーは、最後のキー生成以降に設定可能な時間間隔が経過した後にのみ生成できます。**ssh key-gen wait-time** コマンドは、この待機時間を秒単位で指定します。デフォルトの間隔は 300 秒 (5 分) です。

**ステップ 1** コンテキスト コンフィギュレーション モードを開始します。

```
[local]host_name(config)# context context_name
[local]host_name(config-ctx)#
```

**ステップ 2** 待機時間間隔を指定します。

```
[local]host_name(config-ctx)# ssh key-gen wait-time seconds
[local]host_name(config-ctx)#
```

注：

- *seconds* を 0 ~ 86400 の整数で指定します。デフォルト = 300

## SSH 暗号化暗号方式の指定

SSH コンフィギュレーション モードの **暗号 CLI** コマンドは、SSH 対称暗号化のために、sshd の暗号優先順位リストを設定します。そのコンテキストの暗号オプションが変更されます。

**ステップ 1** SSH コンフィギュレーション モードを開始します。

```
[local]host_name(config-ctx)# server sshd
```

**ステップ 2** 必要な暗号化アルゴリズムを指定します。

```
[local]host_name(config-sshd)# ciphers algorithms
```

注：

- アルゴリズムは 1 ~ 511 文字の英数字の文字列で、次に示すように、優先順位 (左から右) でカンマ区切りの変数 (スペースなし) の単一の文字列として使用するアルゴリズムを指定します。
  - **blowfish-cbc** : 対称キープロック暗号、暗号ブロック連鎖 (CBC)
  - **3des-cbc** : トリプルデータ暗号化規格、CBC
  - **aes128-cbc** : Advanced Encryption Standard (AES; 高度暗号化規格)、128 ビットキーサイズ、CBC
  - **aes128-ctr** : AES、128 ビットキーサイズ、カウンタモード暗号化 (CTR)
  - **aes192-ctr** : AES、192 ビットキーサイズ、CTR
  - **aes256-ctr** : AES、256 ビットキーサイズ、CTR

- **aes128-gcm@openssh.com** : AES、128 ビットキーサイズ、Galois Counter モード [GCM]、OpenSSH
- **aes256-gcm@openssh.com** : AES、256 ビットキーサイズ、GCM、OpenSSH
- **chacha20-poly1305@openssh.com** : ChaCha20 対称暗号、Poly1305 暗号化メッセージ認証コード [MAC]、OpenSSH

通常のビルドにおけるアルゴリズムのデフォルトの文字列は次のとおりです。

```
blowfish-cbc,3des-cbc,aes128-cbc,aes128-ctr,aes192-ctr,aes256-ctr,aes128-gcm@openssh.com,aes256-gcm@openssh.com,
chacha20-poly1305@openssh.com
```

信頼できるビルドにおけるアルゴリズムのデフォルトの文字列は次のとおりです。

```
aes256-ctr,aes192-ctr,aes128-ctr
```

**ステップ 3** SSH コンフィギュレーション モードを終了します。

```
[local]host_name(config-sshd)# end
[local]host_name#
```

## MAC アルゴリズムの設定

### 機能の概要と変更履歴

#### 要約データ

該当製品または機能エリア	すべて
該当プラットフォーム	<ul style="list-style-type: none"> <li>• ASR 5500</li> <li>• VPC-DI</li> <li>• VPC-SI</li> </ul>
機能のデフォルト	無効：設定が必要
このリリースでの関連する変更点	N/A
関連資料	<ul style="list-style-type: none"> <li>• <i>ASR 5500 System Administration Guide</i></li> <li>• <i>Command Line Interface Reference</i></li> <li>• VPC-DI システム管理ガイド</li> <li>• VPC-SI System アドミニストレーションガイド</li> </ul>

#### マニュアルの変更履歴



**重要** リリース 21.2 および N5.1 よりも前に導入された機能の改訂履歴の詳細は示していません。

改訂の詳細	リリース
最初の導入。	21.13

## 機能説明

MAC アルゴリズム設定機能を使用すると、内部 SSHD サーバーの MAC アルゴリズムの優先順位を設定または変更することができます。

この機能をサポートする、新しい CLI **MACs** CLI コマンドが SSH モード設定に導入されました。

## MAC アルゴリズムの設定

ここでは、MAC アルゴリズムの設定方法を説明します。

MAC アルゴリズムの優先順位を指定するには、次の設定を使用します。

```
configure
  context context_name
  server sshd
    macs algorithms
  end

default macs
```

注：

- *algorithms* : 1 ~ 511 文字の英数字文字列を参照します。この文字列は、次のリストで示す優先順位（左から右）のコンマ区切りの変数（スペースなし）の1つの文字列として使用するアルゴリズムを指定します。

- HMAC = ハッシュベースのメッセージ認証コード
- SHA2 = セキュア ハッシュ アルゴリズム 2
- SHA1 = セキュア ハッシュ アルゴリズム 1
- ETM = Encrypt-Then-MAC
- UMAC = ユニバーサルハッシュに基づくメッセージ認証コード

- 次に、通常のビルドのヘルプ文字列とアルゴリズムのリストを示します。

```
hmac-sha2-512-etm@openssh.com, hmac-sha2-256-etm@openssh.com, hmac-sha1-etm@openssh.com, hmac-sha2-512,
hmac-sha2-256, hmac-sha1, urac-128-etm@openssh.com, urac-128@openssh.com, urac-64-etm@openssh.com, urac-64@openssh.com
```

- 次に、信頼できるビルドのヘルプ文字列とアルゴリズムのリストを示します。

```
hmac-sha2-512-etm@openssh.com, hmac-sha2-256-etm@openssh.com, hmac-sha1-etm@openssh.com, hmac-sha2-512,
hmac-sha2-256, hmac-sha1
```

- デフォルト値の文字列は次のとおりです。

```
hmac-sha2-512-etm@openssh.com, hmac-sha2-256-etm@openssh.com, hmac-sha1-etm@openssh.com, hmac-sha2-512,
hmac-sha2-256, hmac-sha1
```

## MAC アルゴリズムの指定

MAC アルゴリズムの優先順位を設定するには、次の CLI コマンドを使用します。このコマンドは、SSH コンフィギュレーションモードで設定します。

```
configure
  context context_name
  server sshd
    macs algorithms
  end
```

**default macs**

注：

- *algorithms* : 1 ~ 511 文字の英数字文字列を参照します。この文字列は、次のリストで示す優先順位（左から右）のコンマ区切りの変数（スペースなし）の1つの文字列として使用するアルゴリズムを指定します。

- HMAC = ハッシュベースのメッセージ認証コード
- SHA2 = セキュア ハッシュ アルゴリズム 2
- SHA1 = セキュア ハッシュ アルゴリズム 1
- ETM = Encrypt-Then-MAC
- UMAC = ユニバーサルハッシュに基づくメッセージ認証コード

- 次に、通常のビルドのヘルプ文字列とアルゴリズムのリストを示します。

```
hmac-sha2-512-etm@openssh.com, hmac-sha2-256-etm@openssh.com, hmac-sha1-etm@openssh.com, hmac-sha2-512,
hmac-sha2-256, hmac-sha1, umac-128-etm@openssh.com, umac-128@openssh.com, umac-64-etm@openssh.com, umac-64@openssh.com
```

- 次に、信頼できるビルドのヘルプ文字列とアルゴリズムのリストを示します。

```
hmac-sha2-512-etm@openssh.com, hmac-sha2-256-etm@openssh.com, hmac-sha1-etm@openssh.com, hmac-sha2-512,
hmac-sha2-256, hmac-sha1
```

- デフォルト値の文字列は次のとおりです。

```
hmac-sha2-512-etm@openssh.com, hmac-sha2-256-etm@openssh.com, hmac-sha1-etm@openssh.com, hmac-sha2-512,
hmac-sha2-256, hmac-sha1
```

## SSH キーの生成

**ssh generate** コマンドは、SSH サーバーによって使用される公開キーまたは秘密キーのペアを生成します。**ssh generate** CLI コマンド内に隠されている **v2-dsa** キーワード。SSH キーを生成するために使用できる唯一のキーワードは、**v2-rsa** です。



**重要** 生成されたキーペアは、コマンドが再度発行されるまで使用中のままになります。

**ステップ 1** コンテキスト コンフィギュレーションモードを開始します。

```
[local]host_name(config)# context context_name
[local]host_name(config-ctx)#
```

ステップ2 SSH キーペアを生成します。

```
[local]host_name(config-ctx)# ssh generate key type v2-rsa
[local]host_name(config-ctx)#
```

## SSH キーペアの設定

**ssh key** コマンドは、システムで使用される公開キーと秘密キーのペアを設定します。**v2-dsa** キーワードは、**ssh key** コマンドでは隠されています。

SSH キーペアのパラメータを指定します。

```
[local]host_name(config-ctx)# ssh key data length octets type v2-rsa
```

注：

- **data** は 1 ～ 1023 文字の英数字の文字列で表される暗号化キーです。
- **length octets** は 0 ～ 65535 の整数で表される、暗号化されたキーのオクテット単位の長さです。
- **type** はキータイプを指定します。**v2-rsa** はサポートされている唯一のタイプです。

**重要** 最大 200 の設定可能な承認済みの SSH キーが StarOS でサポートされています。

## 承認済み SSH ユーザーアクセス

ユーザーが、SSH 認証キーペアを持つ特定のホストから StarOS コンテキストにアクセスすることを許可する必要があります。

### SSH ユーザーアクセスの認可

SSH コンフィギュレーションモードの **authorized-key** コマンドは、指定されたホストからのコンテキストへのユーザーアクセスを許可します。

ステップ1 SSH コンフィギュレーション モードに移動します。

```
[local]host_name(config-ctx)# server sshd
[local]host_name(config-sshd)#
```

ステップ2 **authorized-key** コマンドを使用して管理ユーザーアクセスを指定します。

```
[local]host_name(config-sshd)# authorized-key username user_name host host_ip [ type {
v2-dsa | v2-rsa } ]
```

注：

- **username** *user\_name* は、sshd サーバーへのアクセスに許可されたキーを持つ既存の StarOS 管理者ユーザー名を指定します。*user\_name* は、1～255 文字の英数字文字列で表されます。sshd キーをバイパスしないようにするには、**nopassword** オプションを使用してコンテキストコンフィギュレーションモードの **administrator** コマンドを使用して、ユーザー名を事前に作成しておく必要があります。管理者の作成の詳細については、「システム設定」の章を参照してください。
- **host** *host\_ip* は、このユーザー名の認証キーを持つ SSH ホストの IP アドレスを指定します。この IP アドレスは、IPv4 ドット付き 10 進表記または IPv6 コロン区切り 16 進表記である必要があります。
- **type** はキータイプを指定します。**v2-rsa** はサポートされている唯一のタイプです。

## SSH ユーザーログインの制限事項

管理者は、StarOS CLI への SSH アクセスを、許可されたユーザーの「ホワイトリスト」に制限できます。サービスへのアクセスは、正当なニーズを持つユーザーにのみ制限される場合があります。明示的に許可されたユーザーのみが、SSH を介してホストに接続できます。ユーザー名には、必要に応じて特定の送信元 IP アドレスを含めることができます。

AllowUsers リストは、スペースで区切られたユーザー名パターンで構成されます。パターンで「USER」という形式を使用すると、そのユーザーに対してログインが制限されます。パターンが「USER@IP\_ADDRESS」形式の場合、ユーザーと IP アドレスは個別にチェックされ、指定した IP アドレスからのユーザーへのログインを制限します。

デフォルトでは、任意のユーザーによる無制限のアクセスを許可します。

### 許可済みユーザーリストの作成

**allowusers add** コマンドを使用すると、管理者は StarOS CLI にログインできるユーザーのリストを作成できます。

**ステップ 1** コンテキスト コンフィギュレーション モードを開始します。

```
[local]host_name(config)# context context_name
[local]host_name(config-ctx)#
```

**ステップ 2** SSH コンフィギュレーション モードに移動します。

```
[local]host_name(config-ctx)# server sshd
```

**ステップ 3** SSH ユーザーリストを設定します。

```
[local]host_name(config-sshd)# allowusers add user_list
```

*user\_list* は、スペースで区切られたユーザー名のパターンのリストを、1～999 文字の英数字の文字列として指定します。パターンで「USER」という形式を使用すると、そのユーザーに対してログインが制限されます。

パターンが「USER@IP\_ADDRESS」形式の場合は、ユーザー名と IP アドレスが個別にチェックされ、その特定の IP アドレスからユーザーへのログインが制限されます。

パターンが「USER@<context>@IP\_ADDRESS」形式の場合は、ユーザー名、StarOS コンテキスト、および IP アドレスが個別にチェックされ、その特定の IP アドレスから特定のコンテキストに関連付けられているユーザーへのログインを制限します。

`user_list` には次の制限が適用されます。

- この文字列の最大長は 3000 バイト（スペースを含む）です。
- スペースでカウントされる AllowUsers の最大数は256で、これは OpenSSH からの制限と一致します。

**重要** 上記の制限のいずれかを超えると、エラーメッセージが表示されます。このメッセージでは、正規表現のパターンを使用して文字列を短くするか、または **no allowusers add** や **default allowusers add** を使用してすべての allowusers を削除するか、または再設定するように求められます。

詳細については、『*Command Line Interface Reference*』の「*SSH Configuration Mode Commands*」の章を参照してください。

**ステップ 4** SSH コンフィギュレーション モードを終了します。

```
[local]host_name(config-sshd)# end
[local]host_name#
```

## SSH ユーザーログイン認証

StarOS は、次のシナリオの場合、許可済みキーとユーザーアカウントの組み合わせを使用して SSH によるユーザーログインの試行を認証します。

- ユーザーは、ローカルコンテキスト (VPN) インターフェイスを介してローカルコンテキストのユーザー名と、ローカルコンテキストで設定されている許可済みのキーを使用してログインしようとします。
- ユーザーは、ローカル以外のコンテキストインターフェイスを介してローカル以外のコンテキストのユーザー名と、ローカル以外のコンテキストで設定されている許可済みのキーを使用してログインしようとします。
- ユーザーは、ローカル以外のコンテキストインターフェイスを介してローカルコンテキストのユーザー名と、ローカルコンテキストで設定されている許可済みのキーを使用してログインしようとします。
- ユーザーは、ローカルコンテキストインターフェイスを介してローカル以外のコンテキストのユーザー名と、ローカル以外のコンテキストで設定されている許可済みのキーを使用してログインしようとします。

現在のシステム設定に基づいて認証が失敗すると、ログインが阻止され、エラーメッセージが生成されます。



StarOS では、ユーザー ID が異なるユーザーが同じ公開 SSH キーを使用して、許可されていないコンテキストへログインすることは許可されていません。ユーザーの認証では、許可済みキーとユーザーアカウントの組み合わせが考慮されます。



**重要** StarOS リリース 21.0 以降では、ユーザーがローカル以外のコンテキストからログインした場合、そのユーザーは /flash ディレクトリにアクセスできません。

## セキュアなセッションログアウト

StarOS が SSH クライアントから切断されると、デフォルトの動作によって CLI または SFTP セッションは約 45 秒（デフォルトのパラメータを使用）で終了します。SSH コンフィギュレーションモードの CLI コマンドを使用すると、このデフォルトの SSHD 切断動作を無効にしたり、変更したりできます。



**重要** セキュリティを強化するため、シスコでは、少なくとも `lient-alive-countmax` を 2、`client-alive-interval` を 5 にすることを推奨します。セッションのログアウト値が小さいと、ssh セッションのログアウトが不定期にログアウトする可能性があります。セキュリティとユーザーの使いやすさとのバランスが取れるように値を調整します。

**client-active-countmax** コマンドは、`sshd` なしで送信される `client-alive` メッセージの数を、SSH クライアントからのメッセージを受信しないように設定します（デフォルトは 3）。`client-alive` メッセージの送信中にこのしきい値に達すると、`sshd` は SSH クライアントを切断してセッションを終了します。

**client-alive-interval** コマンドは、タイムアウト間隔を秒単位で設定します（デフォルトは 15）。その後、SSH クライアントからデータを受信しなかった場合、`sshd` は暗号化されたチャネルを介してメッセージを送信し、クライアントからの応答を要求します。メッセージが送信される回数は、`client-alive-countmax` パラメータによって決定されます。`sshd` が SSH クライアントの切断を解除するまでのおおよその時間は、`client-alive-countmax X client-alive-interval` となります。

クライアントまたはサーバーがいつ接続が非アクティブになったかを認識しているかどうかに関係なく、依存している場合、`client-alive` メカニズムは重要です。



**重要** `client-alive` メッセージは暗号化チャネルを介して送信されるため、スプーフィングできません。



**重要** これらのパラメータは、SSH プロトコルバージョン 2 のみに適用されます。

## デフォルトの `sshd` セキュアセッションログアウトパラメータの変更

次のコマンドシーケンスは、クライアントの `ClientAliveCountmax`（デフォルトは 3）および `ClientAliveInterval`（デフォルトは 15 秒）のパラメータのデフォルト設定を変更します。

ステップ 1 コンテキスト コンフィギュレーション モードを開始します。

```
[local]host_name# configure
```

ステップ 2 SSH コンフィギュレーション モードに移動します。

```
[local]host_name(config)# context context_name
```

ステップ 3 `ClientAliveCountmax` パラメータを 2 に設定します。

```
[local]host_name(config-sshd)# client-alive-countmax 2
```

ステップ 4 `ClientAliveInterval` パラメータを 5 秒に設定します。

```
[local]host_name(config-sshd)# client-alive-interval 5
```

ステップ 5 SSH コンフィギュレーション モードを終了します。

```
[local]host_name(config-sshd)# end  
[local]host_name#
```

## 外部サーバーへの SSH クライアントログイン

StarOS は、StarOS ゲートウェイから外部サーバーへの SSH/SFTP アクセスの公開キーの認証をサポートしています。この機能を設定するには、SSH クライアントキーのペアを生成し、クライアント公開キーを外部サーバーにプッシュします。



(注) デフォルトでは、StarOS は外部サーバーへの `username-password` の認証のみをサポートしています。

## SSH クライアント暗号の設定

SSH クライアント コンフィギュレーション モードの `cipher` CLI コマンドは、外部サーバーにログインするときに暗号優先順位リストを設定します。

ステップ 1 SSH クライアント コンフィギュレーション モードを開始します。

```
[local]host_name(config)# client ssh
```

ステップ 2 必要な暗号化アルゴリズムを指定します。

```
[local]host_name(config-ssh)# ciphers algorithms
```

注：

- アルゴリズムは 1 ～ 511 文字の英数字の文字列で、次に示すように、優先順位（左から右）でカンマ区切りの変数（スペースなし）の単一の文字列として使用するアルゴリズムを指定します。
  - **blowfish-cbc**：対称キーブロック暗号、暗号ブロック連鎖（CBC）
  - **3des-cbc**：トリプルデータ暗号化規格、CBC
  - **aes128-cbc**：Advanced Encryption Standard（AES; 高度暗号化規格）、128 ビットキーサイズ、CBC
  - **aes128-ctr**：AES、128 ビットキーサイズ、カウンタモード暗号化（CTR）
  - **aes192-ctr**：AES、192 ビットキーサイズ、CTR
  - **aes256-ctr**：AES、256 ビットキーサイズ、CTR
  - **aes128-gcm@openssh.com**：AES、128 ビットキーサイズ、Galois Counter モード [GCM]、OpenSSH
  - **aes256-gcm@openssh.com**：AES、256 ビットキーサイズ、GCM、OpenSSH
  - **chacha20-poly1305@openssh.com**：ChaCha20 対称暗号、Poly1305 暗号化メッセージ認証コード [MAC]、OpenSSH

通常のビルドにおけるアルゴリズムのデフォルトの文字列は次のとおりです。

```
aes256-ctr,aes192-ctr,aes128-ctr,aes256-gcm@openssh.com,aes128-gcm@openssh.com,chacha20-poly1305@openssh.com,blowfish-cbc,3des-cbc,aes128-cbc
```

信頼できるビルドにおけるアルゴリズムのデフォルトの文字列は次のとおりです。

```
aes256-ctr,aes192-ctr,aes128-ctr
```

**ステップ 3** SSH クライアント コンフィギュレーション モードを終了します。

```
[local]host_name(config-ssh)# end  
[local]host_name#
```

## 優先認証方式の設定

SSH クライアント コンフィギュレーション モードの **preferredauthentications** CLI コマンドは、適切な認証方式を設定します。

**ステップ 1** SSH クライアント コンフィギュレーション モードを開始します。

```
[local]host_name(config)# client ssh
```

**ステップ 2** 優先認証方式の指定

```
[local]host_name(config-ssh)# preferredauthentications methods
```

注：

- 方式：次に示すように、優先順位順（左から右）に、カンマ区切りの変数（スペースなし）の単一の文字列として使用される認証方式を指定します。
  - **publickey**：SSH v2-RSA プロトコルを使用した認証

- **keyboard-interactive** : 任意の数の情報を要求します。各情報について、サーバーはプロンプトのラベルを送信します。
- **password** : 単一のパスワードの単純な要求
- **デフォルト** : 方式の値を [publickey,password] にリセットします。

**ステップ3** SSH クライアント コンフィギュレーション モードを終了します。

```
[local]host_name(config-ssh)# exit
[local]host_name(config)#
```

## SSH クライアントキーペアの生成

SSH クライアント コンフィギュレーション モードでコマンドを使用し、秘密キーを指定して、SSH クライアントキーペアを生成します。

**ステップ1** SSH クライアント コンフィギュレーション モードを開始します。

```
[local]host_name(config)# client ssh
[local]host_name(config-ssh)#
```

**ステップ2** SSH クライアントキーのペアを生成します。

```
[local]host_name(config-ssh)# ssh generate key [ type v2-rsa ] [ key-size ]
[local]host_name(config-ssh)#
```

**type v2-rsa** は SSH クライアントキーのタイプを指定します。サポートされている SSH クライアントキーのタイプは、**v2-rsa** のみです。

**key-size** は SSH クライアントのキーサイズを指定します。サポートされているキーサイズは、2048、3072、4096、5120、6144、7168、および 9216 です。

**ステップ3** SSH クライアントキーが生成されていることを確認します。

```
[local]host_name(config-ssh)# do show ssh client key
```

**ステップ4** SSH クライアント コンフィギュレーション モードを終了します。

```
[local]host_name(config-ssh)# exit
[local]host_name(config)#
```

## 外部サーバーへの SSH クライアント公開キーのプッシュ

このサーバーへの SSH/SFTP アクセスをサポートするには、SSH クライアント公開キーを外部サーバーにプッシュする必要があります。

**ステップ 1** Exec モードで、**push ssh-key** コマンドを実行します。

```
[local]host_name# push ssh-key { host_name | host_ip_address } user username [ context context_name ]
```

```
[local]host_name#
```

**host\_name** は、DNS ルックアップを介して解決される必要がある論理ホスト名を使用してリモートサーバーを指定します。これは、1 ～ 127 文字の英数字文字列で表されます。

**host\_ip\_address** は、IPv4 ドット付き 10 進表記または IPv6 コロン区切り 16 進表記で表されます。

**user username** は、外部サーバーで有効なユーザー名を 1 ～ 79 文字の英数字の文字列として指定します。

**context context\_name** は、有効なコンテキスト名を指定します。コンテキスト名はオプションです。指定されていない場合は、現在のコンテキストが処理に使用されます。

**ステップ 2** 他の外部サーバーでの SSH/SFTP アクセスをサポートするには、ステップ 1 を繰り返します。

**ステップ 3** 外部サーバーへの SSH クライアントのログインをテストします。

```
local]host_name# ssh { hostname | ip_address } user username port port_number
```

## NETCONF の有効化

SSH キーは、NETCONF プロトコルと ConfD エンジンが Cisco Network Service Orchestrator (NSO) をサポートするために有効になる前に必要になります。

NETCONF を有効にする方法の詳細については、このガイドの付録の「NETCONF と ConfD」を参照してください。

## 2 番目の IP アドレスを使用した管理インターフェイスの設定

必要に応じて、vNIC 管理インターフェイスに 2 番目の IP アドレスを設定できます。

### 手順

	コマンドまたはアクション	目的
<b>ステップ 1</b>	プロンプトで次のコマンドを入力して、コンフィギュレーションモードを開始します。	<code>[local]host_name configure</code> <code>[local]host_name(config)</code>
<b>ステップ 2</b>	コンテキスト コンフィギュレーションモードを開始するには、次のように入力します。	<code>[local]host_name(config) context local</code> <code>[local]host-name(config-ctx)</code>

	コマンドまたはアクション	目的
ステップ 3	次のコマンドを使用して、インターフェイスのシロット番号とポート番号を入力します。	<code>[local]host_name(config-ctx) 1/1</code> <code>[local]host_name(config-if-eth)</code>
ステップ 4	次のコマンドを入力して、セカンダリ IP アドレスとサブネットマスクを入力します。	<code>[local]host_name(config-if-eth) { ip   ipv } address ipaddress subnet_mask secondary</code>
ステップ 5	次のコマンドを入力して、コンフィギュレーションモードを終了します。	<code>[local]host_name(config-if-eth) end</code>
ステップ 6	次のコマンドを入力して、インターフェイスの IP アドレスを確認します。	<code>[local]host_name show config context local</code> CLI 出力は次の例のようになります。  <pre>config   context local     interface interface_name       ip address ipaddress subnetmask       ip address ipaddress subnetmask secondary     exit</pre>
ステップ 7	「 <a href="#">インターフェイスとポートの設定の確認と保存 (67 ページ)</a> 」に進みます。	

## Open SSH から Cisco SSH へのアップグレードと移行

### 機能の概要と変更履歴

#### 要約データ

該当製品または機能エリア	すべて
該当プラットフォーム	<ul style="list-style-type: none"> <li>• ASR 5500</li> <li>• VPC-DI</li> <li>• VPC-SI</li> </ul>
機能のデフォルト	有効、常時オン
このリリースでの関連する変更点	N/A
関連資料	<ul style="list-style-type: none"> <li>• <i>ASR 5500 System Administration Guide</i></li> <li>• <i>Command Line Interface Reference</i></li> <li>• <i>VPC-DI システム管理ガイド</i></li> <li>• <i>VPC-SI System アドミニストレーション ガイド</i></li> </ul>

## マニュアルの変更履歴



**重要** リリース 21.2 および N5.1 よりも前に導入された機能の改訂履歴の詳細は示していません。

改訂の詳細	リリース
このリリースでは、暗号と MAC のアルゴリズム値は、OpenSSH から CiscoSSH へのアップグレードと移行に基づいて変更されています。	21.16
最初の導入。	21.2 よりも前

## 変更された機能

Cisco ASR 5500 および VPC 製品のセキュリティ対策として、暗号および MAC アルゴリズム値は、Cisco SSH バージョンへの Open SSH のアップグレードと移行をサポートするように変更されています。

**以前の動作**：21.16 よりも前のリリースでは、**cipher** コマンドと **macs** コマンドの **default** アルゴリズム値は次のようになっていました。

- 暗号化方式

- 21.15 (通常ビルドのみ)

通常のビルドのアルゴリズムの値を次のようにリセットします。

```
blowfish-ctr,3des-ctr,aes128-ctr,aes128-gcm@openssh.com,aes256-gcm@openssh.com,chaCha20-poly1305@openssh.com
```

- MAC

- 21.15 (信頼できるビルドのみ)

信頼できるビルドのアルゴリズムの値を次のようにリセットします。

```
hmac-sha2-512-etm@openssh.com,hmac-sha2-256-etm@openssh.com,hmac-sha1-etm@openssh.com,hmac-sha2-512,hmac-sha2-256,hmac-sha1
```

- KEX アルゴリズム

- 21.15

通常のビルドと信頼できるビルドで使用可能なアルゴリズム：

```
diffie-hellman-group1-sha1,diffie-hellman-group14-sha1
```

**新しい動作**：このリリースでは、**default** コマンドと **cipher** コマンドの **macs** アルゴリズム値は次のとおりです。

- 暗号化方式

リリース 21.16 以降：Post OpenSSH から CiscoSSH へのアップグレードと移行

通常のビルドのデフォルトのアルゴリズムは次のとおりです。

aes256-ctr, aes192-ctr, aes128-ctr, aes256-gcm@openssh.com, aes128-gcm@openssh.com, chacha20-poly1305@openssh.com

通常のビルドで使用可能なアルゴリズムは次のとおりです。

aes256-ctr, aes192-ctr, aes128-ctr, aes256-gcm@openssh.com, aes128-gcm@openssh.com, chacha20-poly1305@openssh.com, aes128-dbc

信頼できるビルドでデフォルトのアルゴリズムと使用可能なアルゴリズム :

aes256-ctr, aes192-ctr, aes128-ctr



- (注) 信頼できるビルドのデフォルトの暗号と設定可能な暗号に変更はありません。

#### • MAC

リリース 21.16 以降 : Post OpenSSH から CiscoSSH へのアップグレードと移行

通常のビルドでデフォルトのアルゴリズムと使用可能なアルゴリズム :

hmac-sha2-512-etm@openssh.com, hmac-sha2-256-etm@openssh.com, hmac-sha1-etm@openssh.com, hmac-sha2-512, hmac-sha2-256, hmac-sha1

信頼できるビルドでデフォルトのアルゴリズム :

hmac-sha2-512, hmac-sha2-256, hmac-sha1

信頼できるビルドで使用可能なアルゴリズム :

hmac-sha2-512, hmac-sha2-256, hmac-sha1



- (注) hmac-sha2-512-etm@openssh.com、hmac-sha2-256-etm@openssh.com、hmac-sha1-etm@openssh.com は信頼できるビルドから削除されます。

#### • KEX アルゴリズム

リリース 21.16 以降 : Post OpenSSH から CiscoSSH へのアップグレードと移行

通常のビルドと信頼できるビルドで使用可能なアルゴリズム :

diffie-hellman-group14-sha1



- (注) KEXアルゴリズムは、StarOSでは設定できません。したがって、CLIの変更はありません。



## VM ハードウェアの検証

リソース割り当ての問題を回避するには、システム内で使用されるすべてのVMが同じサイズのCPUと同じサイズのメモリを持つことが重要です。すべてのインターフェイスでパフォーマンスのバランスを取るために、サービスポートとDIポートが同じスループット能力を備えていることを確認してください。

すべてのカードまたは特定のカードのハードウェア設定を確認するには、**show cloud hardware**`[card_number]` コマンドを使用します。次に、カード1 (CF) でのこのコマンドの出力例を示します。

```
[local]s1# show cloud hardware 1

Card 1:
CPU Nodes           : 1
CPU Cores/Threads   : 8
Memory              : 16384M (qvpc-di-medium)
Hugepage size       : 2048kB
cpeth0              :
  Driver             : virtio_net
loeth0              :
  Driver             : virtio_net
```

次に、カード3 (SF) でのこのコマンドの出力例を示します。

```
[local]s1# show cloud hardware 1

Card 3:
CPU Nodes           : 1
CPU Cores/Threads   : 8
Memory              : 16384M (qvpc-di-medium)
Hugepage size       : 2048kB
cpeth0              :
  Driver             : vmxnet3
port3_10            :
  Driver             : vmxnet3
port3_11            :
  Driver             : vmxnet3
```

基本となるVMハードウェアの最適な設定を表示するには、**show hardware optimum**を使用します。現在のVM設定を最適な設定と比較するには、**show cloud hardware test** コマンドを使用します。最適に設定されていないパラメータは、次の出力例に示すように、アスタリスク付きでフラグが立てられます。この例では、CPUコア/スレッドおよびメモリが最適に設定されていません。

```
[local]s1# show cloud hardware test 1

Card 1:
CPU Nodes           : 1
* CPU Cores/Threads : 8                               Optimum value is 4
* Memory            : 8192M (qvpc-di-medium)         Optimum value is 16384
Hugepage size       : 2048kB
cpeth0              :
  Driver             : virtio_net
loeth0              :
```

```
Driver                : virtio_net
```

設定ディスクまたはローカルフラッシュ上の設定ファイルを表示するには、**show cloud configuration card\_number** コマンドを使用します。フラッシュメモリ上のロケーションパラメータファイルは、インストール時に定義されます。また、ディスク構成は通常、オーケストレーションによって作成され、カードに接続されます。次に、カード1でのこのコマンドの出力例を示します。

```
[local]s1# show cloud configuration 1
```

```
Card 1:
  Config Disk Params:
-----
  No config disk available

  Local Params:
-----
CARDSLOT=1
CARDTYPE=0x40010100
CPUID=0
```

すべてのカードまたは特定のカードの IFTASK 設定を表示するには、**show cloud hardware iftask** コマンドを使用します。デフォルトでは、コアは PMD と VNPU の両方に使用されるように設定されています。次に、カード 4 でのこのコマンドの出力例を示します。

```
[local]mySystem# show cloud hardware iftask 4
Card 4:
  Total number of cores on VM:      24
  Number of cores for PMD only:     0
  Number of cores for VNPU only:    0
  Number of cores for PMD and VNPU: 3
  Number of cores for MCDMA:        4
  Hugepage size:                    2048 kB
  Total hugepages:                   16480256 kB
  NPUSHM hugepages:                  0 kB
  CPU flags: avx sse sse2 ssse3 sse4_1 sse4_2
  Poll CPU's: 1 2 3 4 5 6 7
  KNI reschedule interval: 5 us
```



## 第 4 章

# システム設定

この章では、次の StarOS オプションを設定する手順について説明します。

「使用する前に」で説明したように、システムを最初に設定する手順が完了していることを前提としています。



**重要** この項の設定例で使用されているコマンドは、最もよく使用されるコマンドやキーワードのオプションです。多くの場合は、他のオプションのコマンドやキーワードオプションを使用できます。詳細については、『*Command Line Interface Reference*』を参照してください。

- [インターフェイスとポートの設定の確認と保存 \(67 ページ\)](#)
- [システムタイミングの設定 \(68 ページ\)](#)
- [ソフトウェア RSS の設定 \(73 ページ\)](#)
- [SF ブート設定ポーズの設定 \(74 ページ\)](#)
- [CLI タイムスタンプの有効化 \(75 ページ\)](#)
- [CLI 確認プロンプトの設定 \(75 ページ\)](#)
- [システム管理ユーザーの設定 \(78 ページ\)](#)
- [システム管理ユーザー用の TACACS+ の設定 \(86 ページ\)](#)
- [TACACS+ サーバーでの IPv6 アドレスサポート \(91 ページ\)](#)
- [認証方式の分離 \(91 ページ\)](#)
- [シャーシキーの設定 \(94 ページ\)](#)
- [FSC ファブリックの自動リセットの有効化 \(96 ページ\)](#)

## インターフェイスとポートの設定の確認と保存

次の StarOS CLI コマンドを入力して、インターフェイスの設定が正しいことを確認します。

```
show ip interface
```

このコマンドの出力は、次に示すようになります。この例では、*mangement1* という名前のインターフェイスがローカルコンテキストで設定されています。

```

Intf Name:      LOCAL1
Intf Type:      Broadcast
Description:    management1
VRF:           None
IP State:       UP (Bound to 1/1 untagged, ifIndex 16842753)
IP Address:     192.168.100.3      Subnet Mask:    255.255.255.0
Bcast Address:  192.168.100.255    MTU:           1500
Resoln Type:    ARP              ARP timeout:    60 secs
L3 monitor LC-port switchover: Disabled
Number of Secondary Addresses: 0

```

次のコマンドを入力して、ポートの設定が正しいことを確認します。

```
show configuration port slot/port
```

VPC-SI の場合、スロットは常に 1 です。port はポートの番号 (1, 10 21) です。

この前のコマンドでは、次に示すような出力が生成されます。スロット 1 のポート 1 の設定が表示されます。

```

config
  port ethernet 1/1
    no shutdown
    bind interface LOCAL1 local

```

「設定の確認と保存」の章の説明に従って、設定を保存します。

## システムタイミングの設定

システムには、統計カウンタ、アカウンティングレコード、ロギング、およびイベント通知のタイムスタンプを提供するクロックが備わっています。システムクロックの初期設定後に、1 つまたは複数の Network Time Protocol (NTP) サーバーと通信するようにシステムを設定し、クロックが常に正確であることを確認できます。

すべての VPC インスタンスは、ホストが配置されている IaaS データセンターで使用されているタイミング標準に合わせる必要があります。

タイミングソースの設定に加えて、システムのタイムゾーンを設定する必要があります。

## システムクロックとタイムゾーンの設定

システムクロックとタイムゾーンを設定するには、次のコマンド例を使用します。

```

clock set date:time
configure
  clock timezone timezone [ local ]
end

```

注：

- YYYY:MM:DD:HH:mm 形式 または YYYY:MM:DD:HH:mm:ss に日付、および時刻を入力します。
- サポートされているタイムゾーンの完全なリストについては、**clock timezone** コマンドのオンラインヘルプを参照してください。

- オプションの **local** キーワードは、指定されたタイムゾーンがローカルタイムゾーンであることを示します。
- サマータイムは、そのベースになるタイムゾーンに合わせて自動的に調整されます。

「設定の確認と保存」の章の説明に従って、設定を保存します。

## クロックとタイムゾーンの設定の確認と保存

時間とタイムゾーンが正しく設定されていることを確認するには、次のコマンドを入力します。

```
show clock
```

出力には、設定した日付、時刻、およびタイムゾーンが表示されます。

## ネットワーク時間プロトコルのサポートの設定

この項では、Network Time Protocol (NTP) を使用できるようにするためのシステムの設定に関する情報を提供し、手順を示します。



**重要** NTPサポートを実装する前に、システムクロックとタイムゾーンを設定します。これにより、NTP サーバーで修正する必要がある期間が大幅に短縮されます。



(注) また、NTPはVPC VMを実行している市販（COTS）のすべてのサーバーでも設定する必要があります。StarOSのNTP設定は、COTSサーバーのNTP設定と一致する必要があります。

StarOS が提供するサービスの多くでは、NTP を介して取得する正確な時刻を保持する必要があります。StarOS で使用されている時間基準が正確でないと、サービスの信頼性が失われることがあります。このため、通常のシステム動作には NTP の設定が必要であることを前提とする必要があります。

システムは、NTP を使用して内部クロックを外部の時刻ソース（通常は GPS NTP ソース、あるいはその他の Stratum 2 または 3 サーバー、スイッチ、またはルータ）と同期します。

システムには次が必要です。

- NTP が有効になっていること。
- NTP が、ローカルコンテキスト内のみで使用されるように設定されていること。他のコンテキストを使用すると（enable は設定可能であるため指定できる）、問題が発生します。
- 少なくとも 3 台の外部 NTP サーバー用に NTP が設定されていること。サーバーが 3 台以上ある場合は、サーバーの異常や破損または誤設定が検出され、除外することができます。一般的に、サーバーの数が多ければ（理にかなった範囲で）良好な結果が得られます。



**重要** **prefer** キーワードを使用して外部 NTP サーバーを設定しないでください。NTP クロック選択アルゴリズムには、最適なサーバーを選択する機能がすでに組み込まれています。通常、**prefer** を使用すると、NTP 自体が決定するよりも適切な選択が行われません。



**重要** Cisco TAC から指示された場合を除き、**maxpoll**、**minpoll**、または **version** キーワードの設定は変更しないでください。

必要な NTP 関連付けパラメータを設定するには、次の例を使用します。

```
configure
 ntp
  enable
  server ip_address1
  server ip_address2
  server ip_address3
 end
```

注：

- デフォルトでは *context\_name* は *local* に設定されます。これが推奨されている設定です。
- **server** コマンドには多数のオプションがあります。詳細については、『*Command Line Interface Reference*』の「*NTP Configuration Mode Commands*」の章を参照してください。
- IPv4 のドット付き 10 進表記または IPv6 のコロンで区切られた 16 進表記を使用して、NTP サーバーの IP アドレスを入力します。
- 最大 6 つの NTP サーバー IP アドレスを設定できます。



**重要** 少なくとも 3 台（できれば 4 台）の NTP サーバーを使用してシステムを設定します。

「設定の確認と保存」の章の説明に従って、設定を保存します。

## ローカルソースを使用した NTP サーバーの設定

NTP は、ネットワークピア、ローカルの外部クロック（GPS デバイスなど）、または外部ソースなしのローカルクロックを使用できます。

通常、外部ソースのないローカルクロックは、より適切なクロックが使用できない場合の最終手段です。ほとんどの場合はサイトの間 NTP サーバー上に設定されます。そのため、WAN ネットワークが停止したときに、サイト内のホストは互いに同期し続けることができます。

これは、`ntpd` または多くの市販の NTP デバイスに設定できます。このクロックのストラタム番号は常に上位 (8+) であるため、通常の状態 (実際のソースが使用可能である場合) は、このローカルクロックは使用されません。

## タグ付きインターフェイスでの NTP の設定

タグ付きインターフェイスで NTP を有効にするには、次の NTP コンフィギュレーションモードコマンドを使用します。

```
configure
NTP
  [ no ] vlan vlan_id
end
```

注：

- **vlan *vlan\_id*** : *vlan\_id* は ローカル コンテキスト インターフェイスがバインドされている VLAN です。設定後、NTP デーモンはタグ付きインターフェイスでリッスンを開始します。
- **no vlan** : NTP 設定をデフォルトにリセットし、NTP デーモンがデフォルトのタグなしインターフェイスでリッスンを開始します。

## ロードバランサの使用

NTP デーモンと NTP プロトコルは、設定された各サーバーが NTP を実行していることを前提としています。NTP クライアントが、一連の実際の NTP サーバーにパケットをリレーして配布するロードバランサと同期するように設定されている場合、ロードバランサはそれらのパケットを動的に配布し、NTP クライアントを混乱させる場合があります。NTP パケットは遅延とジッターに敏感です。ロードバランサを介してそれらをリレーすることは、NTP クライアントが混乱する可能性があり、サポートされている操作ではありません。

## NTP の設定確認

NTP 設定が正しいことを確認します。Exec モードのプロンプトで次のコマンドを入力します。

```
show ntp associations
```

出力には、すべての NTP サーバーに関する情報が表示されます。2 つの NTP サーバーを展開する例については、次の出力を参照してください。

```
+----Peer Selection: ( ) - Rejected / No Response
|                    (x) - False Tick
|                    (.) - Excess
|                    (-) - Outlyer
|                    (+) - Candidate
|                    (#) - Selected
|                    (*) - System Peer
|                    (o) - PPS Peer
v
```

```

remote          refid          st t when poll reach  delay  offset jitter
=====
*10.81.254.202  .GPS.          1 u 160 1024 377   21.516  0.019  0.009

```

次の表では、**show ntp association** コマンドによって出力されるパラメータについて記述しています。

表 9: NTP パラメータ

列のタイトル	説明
リモート	現在の NTP サーバーのリスト。これらの文字の 1 つは、各 IP アドレスの前にあり、サーバーの現在の状態を示します。 <ul style="list-style-type: none"> <li>• ( ) Rejected/No response</li> <li>• X False tick</li> <li>• . Excess</li> <li>• - Outlyer</li> <li>• + Candidate</li> <li>• # Selected</li> <li>• * System peer</li> <li>• (o) PPS peer</li> </ul>
refid	サーバーが同期している最後に報告された NTP の参照。
st	NTP サーバーのストラタムレベル。
t	通信タイプ：ブロードキャスト、マルチキャストなど
when	最終接続からの経過秒数。
poll	システムと NTP サーバー間の電話機の設定ユーティリティ。
reach	この NTP サーバーへの過去 8 回のポーリングで受信された応答を示す、到達可能性シフトレジスタの 8 進値。
delay	システムと NTP サーバー間で交換されるメッセージのラウンドトリップ遅延（ミリ秒単位）。
offset	システムクロックを NTP サーバーと同期させるために調整する必要があるミリ秒数。
ジッター	システムと NTP サーバー間のジッター（ミリ秒単位）。



## ソフトウェア RSS の設定

Cisco Unified Computing System (USC) の NIC は、ハードウェアベースの受信側スケーリング (RSS) をサポートしています。ただし、RSS は IP トラフィックでのみサポートされています。MPLS、GTP、L2TP、GREなどの他のネットワークプロトコルでは、すべてのトラフィックが単一のキューにルーティングされます。

VPC-SI は、処理のために使用可能な vCPU コアに MPLS トラフィックを配信するソフトウェア RSS 機能を提供します。これにより、リソース使用率が増加し、スループットが向上します。

ソフトウェア RSS 機能は、Cisco UCS NIC ハードウェア RSS サポートを補足することができます。これは、ハードウェア NIC でサポートされていない一部のトラフィック (このリリースでのみ MPLS トラフィック) を配信することを意味します。VPC-SI は、包括的な RSS カバレッジを提供することもできます。つまり、すべてのトラフィックを配信します。このオプションは、RSS をサポートしていないハードウェアが使用されている場合に適用されます。

`iftask sw-rss` コマンドで RSS を使用するように設定します。

```
config
  iftask sw-rss {comprehensive | supplemental}
```

すべての着信トラフィックに対して RSS を設定するには、**comprehensive** キーワードを使用します。**supplemental** キーワードを使用して、ハードウェア RSS 機能でサポートされていないプロトコル (このリリースでのみ MPLS トラフィック) で RSS を設定します。

## DI ネットワーク RSS の暗号化

### 機能の概要と変更履歴

#### 要約データ

該当製品または機能エリア	すべて
該当プラットフォーム	VPC-DI
機能のデフォルト	無効：設定が必要
このリリースでの関連する変更点	N/A
関連資料	VPC-DI システム管理ガイド

## マニュアルの変更履歴



**重要** リリース 21.2 および N5.1 よりも前に導入された機能の改訂履歴の詳細は示していません。

改訂の詳細	リリース
Distributed Instance Network (DI ネットワーク) の RSS トラフィックのデフォルト設定は現在は無効になっていますが、新しい CLI コマンドを使用して有効にすることができます。以前のリリースでは、この機能は自動的に有効になっており、設定できませんでした。	21.8
最初の導入。	21.2 よりも前

## 変更された機能

**以前の動作** : 21.8 よりも前のリリースでは、仮想化 StarOS インスタンスの内部分散インスタンスネットワーク (DI ネットワーク) 上のすべてのトラフィックに対して **Receive Side Scaling (RSS)** がデフォルトで有効になっていました。

**新しい動作** : リリース 21.8 以降では、RSS はデフォルトで無効になっており、新しい CLI を介して有効にすることができます。

## コマンドの変更

### iftask di-net-encrypt-rss

この新しい CLI コマンドは、DI ネットワーク上の暗号化トラフィックでの RSS の有効化を制御するために追加されました。

```
configure
  [no] iftask di-net-encrypt-rss
end
```



(注) デフォルト設定では無効になっています。

## SF ブート設定ポーズの設定

特定の状況下では、VPC-DI 展開内で、すべての SF が起動プロセスを完了する前に CF がブート設定を適用します。

次のコンフィギュレーションモードコマンドを実行すると、**wait cards active** 指定されたすべてのカードが動作可能になるか、タイムアウト期間が経過するまで (いずれかの基準が最初に満たされるまで) 設定が一時停止されます。一時停止は、ローカル管理コンテキストの作成と ntp/snmp 設定の直後に行われます。

このコマンドは、シャーシのロードまたはリロード後に SF がオンラインになっており、それらの SF に関する設定が適用されない（その結果、失われる）シナリオを修正します。

#### configure

```
[ no ] wait cards active { all | number } [ standby number ] timeout  
seconds  
end
```

注：

- **all**：すべてのアクティブモードカードが動作ステータスを達成するまで一時停止します。
- **number**：指定した数のアクティブモードカードが動作ステータスになるまで一時停止します。**number** は、アクティブモードカードの数によって 0 になります。
- **standby number**：（オプション）指定した数の非アクティブモードカードが動作ステータスになるまで待機します。

**number** は、0 またはアクティブモード SF 用に設定されていないサービススロットの数です。

- **timeout seconds**：指定されたカードセットが動作ステータスになるまで、1～3600 秒待ちます。この状態が満たされた場合に、待機が終了します。それ以外の場合は、タイムアウト期間が経過すると待機が終了します。

次のコマンド例では、すべてのアクティブカードと1つのスタンバイカードがアクティブになるまで、最大 120 秒待機するようにシステムに指示します。

```
wait cards active all standby 1 timeout 120
```

## CLI タイムスタンプの有効化

CLI で実行されるすべてのコマンドのタイムスタンプ（日付と時刻）を表示するには、Exec モードのルートプロンプトで次のコマンドを入力します。

#### timestamps

日付と時刻は、このコマンドを実行するとすぐに表示されます。

「設定の確認と保存」の章の説明に従って、設定を保存します。

## CLI 確認プロンプトの設定

Exec モードとグローバル コンフィギュレーション モードのさまざまなコマンドによって、コマンドを実行する前にユーザーに確認のプロンプト（Are you sure? [Yes|No]:）が表示されません。

この項では、次の設定について説明します。

- 現在の CLI セッション (Exec モード) またはすべての CLI セッションとユーザー (グローバル コンフィギュレーション モード) のコマンドを自動的に確認する。
- Exec モードの **configure** コマンドと **autoconfirm** コマンドに対してのみ確認を要求する。
- Exec モード設定コマンドの確認を選択的に要求する。

## 自動確認の有効化

**autoconfirm** コマンドを使用して、設定コマンドの確認プロンプトを無効にできます。**autoconfirm** コマンドは、Exec モードおよびグローバル コンフィギュレーション モードで使用できます。自動確認機能を有効にすると、設定コマンドプロンプトに「Yes」の応答が自動的に提供されます。これには、リロードやシャットダウンなどの重要なコマンドが含まれます。デフォルトでは自動確認機能は無効になっています。

Exec モードでは、自動確認は現在のインタラクティブ CLI セッションにのみ適用されます。

グローバル コンフィギュレーション モードでは、すべての CLI ユーザーのすべての CLI セッションに自動確認が適用されます。

```
configure
  autoconfirm
end
```

自動確認を有効にした後に無効にするには、**no autoconfirm** コマンドを使用します。




---

**重要** コマンドガードが有効になっている場合、自動確認はコマンドガードを無効にします。

---

自動確認は「ease-of-use」機能として意図されています。「Are you sure?」という答えがあることを前提としています。[Y/N] プロンプトは「Yes」になり、プロンプトをスキップします。この使用は、ユーザーがこれらの「safety-net」プロンプトを必要としないエキスパートであることを意味します。

## autoconfirm コマンドと configure コマンドに対する確認の要求

**autoconfirm** (Exec モードとグローバル コンフィギュレーションモード) コマンドと、グローバル コンフィギュレーションモードの **commandguard** コマンドを介した **configure** (Exec モード) コマンドに対する確認プロンプトが必要な場合があります。




---

**重要** **autoconfirm** が有効になっている場合、Exec モードとグローバル コンフィギュレーションモードの両方で **autoconfirm** が無効になるまで **commandguard** は有効になりません。

---

次のコマンドシーケンスは、**commandguard** 機能を有効にします。

```
configure
  commandguard
end
```

`commandguard` を有効にすると、次の例に示すように確認プロンプトが表示されます。

```
[local]host_name# configure
Are you sure? [Yes|No]: yes
[local]host_name(config)#
```

`commandguard` がいったん有効になった後に無効にするには、**no commandguard** コマンドを使用します。

`commandguard` のステータスは、**show configuration** コマンドで出力されます。

## 特定の Exec モードのコマンドに対する確認の要求

`commandguard` コマンドのキーワードを使用すると、自動確認が有効になっている場合でも、Exec モード コンフィギュレーション コマンドの特定のカテゴリに対して必須のプロンプトを適用できます。

コマンド構文は次のとおりです。

```
configure
  commandguard exec-command exec_mode_category
end
```

注：

- **exec-command** *exec\_mode\_category* は、Exec モード コンフィギュレーション コマンドのいずれかのカテゴリを指定します。
  - card
  - clear
  - copy
  - debug
  - 削除
  - filesystem
  - hd
  - reload
  - rename
  - shutdown
  - task
  - upgrade
- 複数の **commandguard exec-command** *exec\_mode\_category* コマンドを入力できます。
- 自動確認が有効になっているかどうかに関係なく、指定したカテゴリワードで始まる Exec モードのすべてのコマンドに確認のプロンプトを表示します。
- **no commandguard exec-command** *exec\_mode\_category* コマンドを使用して、特定のカテゴリの確認プロンプトをオフにすることができます。
- Exec モードコマンドに対する **commandguard exec-command** で自動確認をオーバーライドした場合、StarOS はコマンドを実行しようとしたときに自動確認がオーバーライドされた理由を示す情報メッセージを表示します。

- **-noconfirm** キーワードをサポートする Exec モード コンフィギュレーション コマンドに対する確認プロンプトのオーバーライドを選択できます。

たとえば、**commandguard exec-command card** が有効になっている場合、確認プロンプトは次のように表示されます。

```
[local]host_name# card busy-out 1
Info: commandguard prevents autoconfirm of this command
Are you sure? [Yes|No]: yes
[local]host_name#
```

## システム管理ユーザーの設定

はじめに、システムのコンテキストレベルのセキュリティ管理者を設定する方法について説明します。

この項では、次の権限を持つ追加の管理ユーザーを設定する手順について説明します。

- **セキュリティ管理者**：読み取り/書き込み権限を持ち、管理者、演算子、およびインスペクターで使用できるものを含め、すべての CLI コマンドを実行できます。
- **管理者**：読み取り/書き込み権限を持ち、CLI で任意のコマンドを実行できます。ただし、セキュリティ管理者だけが設定できるセキュリティ関連のコマンドは実行できません。管理者は、システム設定を構成または変更したり、オペレータやインスペクターで使用可能なコマンドを含め、すべてのシステムコマンドを実行したりできます。
- **オペレータ**：Exec モードコマンドのより大きなサブセットに対する読み取り専用権限を持ちます。これらは、インスペクタモードの一部であるすべてのコマンドと、システムモニタリング、統計情報、および障害管理機能の一部を実行できます。オペレータには、コンフィギュレーションモードを開始する機能がありません。
- **インスペクタ**：いくつかの読み取り専用 Exec モードコマンドに制限されます。これらのほとんどは、さまざまな統計情報や条件を表示するための **show** コマンドです。インスペクタは、**show configuration** コマンドを実行できず、設定モードを開始する権限もありません。

設定手順は、管理ユーザーのタイプ（コンテキストレベルまたはローカルユーザー）に従って分類されます。



---

**重要** これらのユーザー権限とタイプの違いについては、「はじめに」を参照してください。

---

## ユーザー名の文字制限

ユーザー名には、英数字 (a-z、A-Z、0-9)、ハイフン、下線、およびピリオドだけを含めることができます。ハイフン文字を最初の文字にすることはできません。これは、AAA ユーザー名とローカルユーザー名に適用されます。

これらの標準規格に準拠していないユーザー名を作成しようとすると、「Invalid character」というメッセージが表示されます。有効な文字「0123456789.-\_abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ」です。

## コンテキストレベルの管理ユーザーの設定

このユーザータイプはコンテキストレベルで設定され、ログイン時にユーザー名とパスワードを検証するために AAA サブシステムに依存します。これは、設定ファイルを介してローカルに設定された管理ユーザーアカウント、または外部 RADIUS サーバーまたは TACACS+ サーバーの両方に当てはまります。これらのユーザータイプのパスワードは一度割り当てられ、設定ファイルでアクセスできます。

この項では、コンテキストレベルの管理ユーザータイプを設定するための情報と手順について説明します。

アカウントごとまたは認証方式ごとにシミュレーションの CLI セッションの最大数を設定することができます。セキュリティ設定と属性に影響を与えたり、システムのサービス、安定性、およびパフォーマンスに悪影響を及ぼす可能性がある特定のアカウントを保護します。同時 CLI セッションの最大数は、新しいローカルユーザーログインと新しい AAA コンテキストベースのログインを試行するときに設定できます。セッションの最大数が 0 に設定されている場合は、ログインタイプに関係なくユーザーが認証されます。CLI タスクが開始されると、カウントを識別するためのチェックが完了します。この場合、CLI は、そのユーザーのセッション数が 0 よりも大きい 1 であると判断し、出力にエラーメッセージを表示し、`starCLIActiveCount` オブジェクトおよび `starCLIMaxCount` SNMP MIB オブジェクトと、`starGlobalCLISessionsLimit` アラームおよび `starUserCLISessionsLimit` SNMP MIB アラームを生成します。

ローカルユーザーのユーザー名におけるグローバル コンフィギュレーション モードのコマンドの `max sessions` キーワードは、ローカルユーザーに使用可能な同時セッションの最大数を設定します。

最大セッションにおけるコンテキスト コンフィギュレーション モードのコマンドを使用すると、管理ユーザーは、対応するユーザーに許可される最大同時セッション数を設定できます。

これらのコマンドの詳細については、『*Command Line Interface Reference*』を参照してください。

### 警告期間中のパスワード変更オプション

警告期間中でも、パスワードを変更できます。たとえば、次にサンプル出力を示します。

```
When in warning period
1.Warning: Your password is about to expire in 3 days.
   We recommend you to change password.
   Logins are not allowed without acknowledging this.
   Do you want to change it now ? [y/n] (times out in 30 seconds) : n
# <you are logged in>
2.Warning: Your password is about to expire in 3 days.
   We recommend you to change password.
   Logins are not allowed without acknowledging this.
   Do you want to change it now ? [y/n] (times out in 30 seconds) : y
Auto generated password : <Jc42Q8hU>
Do you want to use auto-generated password? [y/n]: n
New password:
```

```
Repeat new password:
# <you are logged in>
```

## コンテキストレベルのセキュリティ管理者の設定

追加のセキュリティ管理者を設定するには、次の例を使用します。

```
configure
  context local
    administrator user_name { [ encrypted ] [ nopassword ] password password
  }
  end
```

注：

- 追加のキーワードオプションを使用して、アクティブな管理者を識別したり、管理者に時間しきい値を設定したりできます。**administrator** コマンドの詳細については、『*Command Line Interface Reference*』を参照してください。
- **nopassword** オプションを使用すると、関連付けられたパスワードを使用せずに管理者を作成できます。認証の唯一の手段として ssh 公開キー (SSH コンフィギュレーションモードの **authorized key** コマンド) を使用する場合は、このオプションを有効にします。このオプションを有効にすると、管理者パスワードを使用してユーザーアカウントにアクセスできなくなります。

「設定の確認と保存」の章の説明に従って、設定を保存します。

## コンテキストレベルの管理者の設定

次の例を使用して、コンテキストレベルの設定管理者を設定します。

```
configure
  context local
    config-administrator user_name { [ encrypted ] [ nopassword ] password
  password }
  end
```

注：

- 追加のキーワードオプションを使用して、アクティブな管理者を識別したり、管理者に時間しきい値を設定したりできます。**config-administrator** コマンドの詳細については、『*Command Line Interface Reference*』を参照してください。
- **nopassword** オプションを使用すると、関連付けられたパスワードを使用せずに設定管理者を作成できます。認証の唯一の手段として ssh 公開キー (SSH コンフィギュレーションモードの **authorized key** コマンド) を使用する場合は、このオプションを有効にします。このオプションを有効にすると、設定管理者パスワードを使用してユーザーアカウントにアクセスできなくなります。

「設定の確認と保存」の章の説明に従って、設定を保存します。

## コンテキストレベルのオペレータの設定

次の例を使用して、コンテキストレベルのオペレータを設定します。



```
configure
  context local
    operator user_name { [ encrypted ] [ nopassword ] password password }
  end
```

注：

- 追加のキーワードオプションを使用して、アクティブな管理者を識別したり、管理者に時間しきい値を設定したりできます。オペレータコマンドの詳細については、『**Command Line Interface Reference**』を参照してください。
- **nopassword** オプションを使用すると、関連付けられたパスワードなしでオペレータを作成できます。認証の唯一の手段として ssh 公開キー（SSH コンフィギュレーションモードの **authorized key** コマンド）を使用する場合は、このオプションを有効にします。このオプションを有効にすると、オペレータパスワードを使用してユーザーアカウントにアクセスできなくなります。

「設定の確認と保存」の章の説明に従って、設定を保存します。

## コンテキストレベルのインスペクタの設定

次の例を使用して、コンテキストレベルのインスペクタを設定します。

```
configure
  context local
    inspector user_name { [ encrypted ] [ nopassword ] password password
  }
  end
```

注：

- 追加のキーワードオプションを使用して、アクティブな管理者を識別したり、管理者に時間しきい値を設定したりできます。**Inspector** コマンドの詳細については、『*CommandLine Interface Reference*』を参照してください。
- **nopassword** オプションを使用すると、関連付けられたパスワードを使用することなくインスペクタを作成できます。認証の唯一の手段として ssh 公開キー（SSH コンフィギュレーションモードの **authorized key** コマンド）を使用する場合は、このオプションを有効にします。このオプションを有効にすると、インスペクタパスワードを使用してユーザーアカウントにアクセスできなくなります。

「設定の確認と保存」の章の説明に従って、設定を保存します。

## システムと LI 設定の分離

合法的傍受（LI）の設定には、機密情報が含まれています。通常のビルドのデフォルトでは、**li-administration** 権限のない管理者が LI 設定コマンドを表示できます。ただし、LI 設定コマンドの表示は、システム設定の残りの部分から制限したり、分離したりすることができます。

グローバル コンフィギュレーション モードの **require segregated li-configuration** コマンドによって、システムおよび合法的傍受の CLI の表示は永続的に分離されます。合法的傍受のキーワードを指定した CLI コマンドは暗号化され、**li-administration** 権限を持つ管理者によってのみ表示できます。



**重要** 信頼できるビルドでは、LIの分離はオンになっており、無効にすることはできません。**require segregated li-configuration** コマンドは非表示です。

システム設定から LI 設定を分離すると、StarOS に次の影響があります。

- **li-administration** 権限を持つ管理者のみが、**show configuration** コマンドの出力で合法的傍受の CLI コマンドを確認できます。
- **save configuration** コマンドを実行すると、合法的傍受の CLI 設定コマンドが自動的に暗号化されます。
- CLI コマンドを介して保存されたコンフィギュレーションファイルをロードすると（たとえば、**configure <url>**）、暗号化済み合法的傍受の CLI コマンドは、LI 権限を持つ管理者に対してのみ復号および実行されます。LI 権限のない管理者の場合は、暗号化済み合法的傍受の CLI コマンドは復号および実行されません。
- 起動設定がロードされているシステム起動中に、暗号化済み合法的傍受の設定が復号され、サイレントにロードされます。つまり、合法的傍受の CLI 設定はコンソールポートに表示されません。
- Exec モードの **configure** コマンドは、LI 管理者が保存されたコンフィギュレーションファイルから、暗号化された合法的傍受の設定のみをロードできるようにするキーワードをサポートするようになりました（たとえば、**configure encrypted <url>** など）。暗号化されたキーワードは、LI 管理者のみが実行できます。
- 暗号化済み合法的傍受の設定（分離された LI）を使用してシステムを実行している場合、**show boot initial config** コマンドの出力には、最初の起動時に 2 番目のパスを実行する必要があるかどうかを示す行が含まれています。この行には、暗号化済み合法的傍受の設定が処理された場合は「**encrypted li**」と表示されます。その行に「**encrypted li errors**」と表示されている場合は、2 番目のパスが正しく動作しなかったか、予期されていないまたは本来の情報ではない出力が提供されています。
- **li-administration** 権限を持つユーザーは、**show logs encrypted-li** コマンドを使用して、暗号化済み合法的傍受の設定におけるブート設定の出力を表示できます。

グローバル コンフィギュレーション モードの **require segregated li-configuration** コマンドと関連コマンドの詳細については、『*Lawful Intercept Configuration Guide*』の付録「*Lawful Intercept CLI Commands*」を参照してください。



(注) 『*Lawful Intercept Configuration Guide*』は、[www.cisco.com](http://www.cisco.com) では利用できません。このガイドのコピーを入手するには、シスコのアカウント担当者にお問い合わせください。

リリース 21.4 以降（信頼できるビルドのみ）：

- ユーザーは、それぞれのコンテキストインターフェイスを使用してのみシステムにアクセスできます。

- ユーザーが別のコンテキストインターフェイスを使用してそれぞれのコンテキストにログインしようとする、そのユーザーは拒否されます。
- ユーザーが「authorized-keys」または「allowusers」のいずれかのコンテキストに設定されているかどうかに関係なく、この機能では、独自のコンテキストインターフェイス以外のコンテキストインターフェイスを介してログインを試行すると、これらのユーザーは拒否されます。
- ローカル以外のコンテキストに設定されているユーザーは、ログインを試行するコンテキストを指定する必要があります。次に例を示します。

```
ssh username@ctx_name@ctx_ip_addr
```

## コンテキストレベルの管理ユーザー設定の確認

設定が正常に行われたことを確認するには、次のコマンドを入力します。

```
show configuration context local
```

このコマンドは、このセッション時にローカルコンテキスト内で変更したすべての設定パラメータを表示します。次に、このコマンドの出力例を示します。この例では、*testadmin* という名前のセキュリティ管理者が設定されています。

```
config
context local
  interface mgmt1
    ip address 192.168.1.10 255.255.255.0
  #exit
  subscriber default
  #exit
  administrator testadmin encrypted password fd01268373c5da85
  inspector testinspector encrypted password 148661a0bb12cd59
exit
port ethernet 5/1
  bind interface mgmt1 local
#exit
```

## ローカルユーザーの管理ユーザーの設定

ローカルユーザータイプは、ANSI T1.276-2003 のパスワードセキュリティ保護をサポートしています。パスワード、パスワード履歴、ロックアウト状態などのローカルユーザーアカウント情報は、/flash で維持されます。この情報は、AAA ベースの認証の対象となる個別のローカルユーザーデータベースにただちに保存され、システムの残りの部分では使用されません。そのため、設定されているローカルユーザーアカウントは、システム設定の残りの部分では表示されません。



---

**重要** ローカルユーザーデータベースが無効になっています。グローバルコンフィギュレーションモードの **local-user** コマンドと Exec モードの **show local-user** コマンド、および **update local-user** コマンドは使用できません。信頼できるビルドの詳細については、「システム操作と設定」の章を参照してください。

---

次の例を使用して、ローカルユーザーの管理ユーザーを設定します。

```
configure
  local-user username name
end
```

注：

- 追加のキーワードオプションを使用して、アクティブな管理者を特定したり、管理者に時間のしきい値を設定したりできます。*local-user username* コマンドの詳細については、『**Command Line Interface Reference**』を参照してください。

ローカルユーザーデータベースの詳細については、[ローカルユーザーデータベースの更新とダウングレード \(84 ページ\)](#) を参照してください。

## ローカルユーザー設定の確認

設定が正常に行われたことを確認するには、次のコマンドを入力します。

```
show local-user verbose
```

このコマンドは、設定されたローカルユーザーの管理ユーザーに関する情報を表示します。次に、このコマンドの出力例を示します。この例では、*SAUser* という名前のローカルユーザーが設定されています。

```
Username:                SAUser
Auth Level:              secadmin
Last Login:              Never
Login Failures:          0
Password Expired:        Yes
Locked:                  No
Suspended:               No
Lockout on Pw Aging:     Yes
Lockout on Login Fail:   Yes
```

## ローカルユーザーデータベースの更新

次の Exec モードコマンドを実行して、ローカルユーザー（管理）設定を更新します。このコマンドは、管理ユーザーを作成、削除、または編集した直後に実行する必要があります。

```
update local-user database
```

## ローカルユーザーデータベースの更新とダウングレード

PBKDF2（パスワードベースのキー派生関数バージョン2）を使用し、入力されたデータ、ソルト、および反復回数に基づいて、特定の長さのキーを取得します。ローカルユーザーアカウントのパスワードは、ランダムに生成されたソルトと多くの反復回数を備えた PBKDF2 方式を使用してハッシュされ、パスワードストレージの保護が強化されます。

リリース 20.0 にアップグレードする場合、ローカルユーザーデータベース内の既存のユーザーパスワードは MD5 から PBKDF2 ハッシングに自動的にアップグレードされません（ハッシュされたパスワード値のみが保存されます）。ハッシュ関数は一方向であるため、保存されたハッシュ値からユーザーパスワードを取得することはできません。したがって、既存のハッシュパスワードを強力なハッシュパスワードに自動的に変換することはできません。

データベースを更新するには、セキュリティ管理者が Exec モードの **update local-user database** CLI コマンドを実行する必要があります。このコマンドを実行すると、StarOS は /flash ディレクトリからデータベースを読み取り、データベースを新しい形式で再構築してディスクに書き込みます。

一時停止されたユーザーを再アクティブ化するには、セキュリティ管理者が次の手順を実行します。

- Exec モードの **password change local-user username** コマンドを使用して、一時停止されたユーザーの一時パスワードを設定します。
- コンフィギュレーション モードの **no suspend local-user username** コマンドを使用して、ユーザーの一時停止フラグをリセットします。

## 指定されたルートディレクトリへのユーザーアクセスの制限

デフォルトでは、FTP/SFTP アクセス権を持つ管理者ユーザーは、/mnt/user/ ディレクトリの下にある任意のファイルにアクセスし、変更を加えることができます。次のディレクトリに「オールオアナッシング」ベースでアクセスが許可されます。/flash、/cdrom、/hdraid、/records、/usb1、/usb2

管理者または設定管理者は、ファイルディレクトリとアクセス権限を使用して SFTP サブシステムのリストを作成できます。ローカルユーザーが作成されると、管理者は SFTP サブシステムを割り当てます。ユーザーの承認レベルがセキュリティ管理者または管理者でない場合、ユーザーは読み取り専用権限を持つサブシステムにのみアクセスできます。このディレクトリは、ユーザーのルートディレクトリとして使用されます。この情報は、`openssh sftp` サーバーに渡される環境変数として設定されます。

SFTP ルートディレクトリは、ローカルユーザー、管理者、および設定管理者と関連付ける前に作成する必要があります。複数の SFTP ディレクトリを作成できます。各ディレクトリは、1 人以上のユーザーに割り当てることができます。

## SFTP ルートディレクトリの設定

`subsystem sftp` コマンドを使用すると、SFTP ルートディレクトリおよび関連付けられているアクセス権限レベルを割り当てることができます。

```
configure
  context local
    server sshd
      subsystem sftp [ name sftp_name root-dir pathname mode { read-only
| readwrite } ]
```

注：

- *sftp\_name* は、このサブシステムを一意に識別する英数字の文字列です。
- *pathname* は、SFTP ファイルの転送先となるルートディレクトリを指定します。次のオプションがあります。
  - /hd-raid/records/cdr

- /flash

## SFTP ルートディレクトリとローカルユーザーの関連付け

**local-user username** コマンドを使用すると、管理者は SFTP ルートディレクトリを指定されたユーザー名に関連付けることができます。

```
configure
  local-user username user_name authorization-level level ftp sftp-server
  sftp_name password password
  exit
```

## SFTP ルートディレクトリと管理者の関連付け

**administrator** コマンドを使用すると、管理者は、指定された管理者に SFTP ルートディレクトリを関連付けることができます。

```
configure
  context local
  administrator user_name password password ftp sftp-server sftp_name
  exit
```

## SFTP ルートディレクトリと設定管理者の関連付け

**config-administrator** コマンドを使用すると、管理者は、指定された設定管理者に SFTP ルートディレクトリを関連付けることができます。

```
configure
  context local
  config-administrator user_name password password ftp sftp-server sftp_name
  exit
```

# システム管理ユーザー用の TACACS+ の設定

ここでは、VPC-SI の TACACS+ (Terminal Access Controller Access Control System+) AAA (認証、許可、およびアカウントリング) サービスの機能と設定について説明します。

## 動作

TACACS+は、セキュアな暗号化されたプロトコルです。管理ユーザーアカウントデータベースを使用してプロビジョニングされた TACACS+ サーバーにリモートでアクセスすることで、VPC-SI システムはシステム管理ユーザーに TACACS+ AAA サービスを提供できます。TACACS+ は、UDP ではなく TCP を使用する TACACS プロトコルの拡張バージョンです。

システムは、TACACS+ ネットワークアクセスサーバー (NAS) として機能します。NAS として、承認されたシステム管理ユーザーに代わって、システムが TACACS+ AAA サービスを要求します。正常に認証されるには、システムがアクセスするのと同じローカルコンテキストとネットワーク内に TACACS+ サーバーが存在する必要があります。

システムは、TACACS+マルチ接続モードをサポートしています。マルチ接続モードでは、セッションごとに TACACS+ サーバーへの個別のプライベート TCP 接続が開き、維持されます。TACACS+ セッションが終了すると、サーバーへの接続が終了します。

TACACS+ は、VPC-SI 上のシステム全体を網羅する機能です。TACACS+ AAA サービスの設定は、TACACS コンフィギュレーションモードで実行されます。TACACS+ 機能の有効化は、グローバル コンフィギュレーションモードで実行されます。システムでは、最大 3 台の TACACS+ サーバーの設定がサポートされます。

システムで設定して有効にすると、TACACS+ 認証が最初に試行されます。デフォルトでは、TACACS+ 認証が失敗すると、システムは RADIUS などの TACACS+ 以外の AAA サービスを使用してユーザーの認証を試行します。

アカウントごとまたは認証方式ごとにシミュレーションの CLI セッションの最大数を設定することができます。セキュリティ設定と属性に影響を与えたり、システムのサービス、安定性、およびパフォーマンスに悪影響を及ぼす可能性がある特定のアカウントを保護します。同時 CLI セッションの最大数は、新しい TACACS+ ユーザーログインを試行するときに設定できます。最大セッション機能を使用するには、TACACS+ サーバー属性オプションのオプションを使用することを **推奨** します。2 番目の方法は、**user-id** コマンドに **maxsess** キーワードを指定して StarOS CLI コンフィギュレーションモードの TACACS+ モードを使用することです。セッションの最大数が 0 に設定されている場合は、ログインタイプに関係なくユーザーが認証されます。CLI タスクが開始されると、カウントを識別するためのチェックが完了します。この場合、CLI は、そのユーザーのセッション数が 0 よりも大きい 1 であると判断し、出力にエラーメッセージを表示し、**starCLIActiveCount** オブジェクトおよび **starCLIMaxCount** SNMP MIB オブジェクトと、**starGlobalCLISessionsLimit** アラームおよび **starUserCLISessionsLimit** SNMP MIB アラームを生成します。

**max-sessions** TACACS+ コンフィギュレーション モード コマンドは、TACACS+ で使用可能なセッションの最大数を設定します。また、**user-id** TACACS+ コンフィギュレーション モード コマンドの **default** オプションは、特定の TACACS+ ユーザー識別子にデフォルトの属性を設定します。これらのコマンドの詳細については、『*Command Line Interface Reference*』を参照してください。



---

**重要** ユーザーは、StarOS と TACACS+ サーバーの両方の設定で使用可能なシミュレーション CLI セッションの最大数を定義できます。ただし、このオプションを使用しないことを強く推奨します。

---



---

**重要** TACACS+ アカウンティング (CLI イベントロギング) は、権限レベルが 15 と 13 に設定されている合法的傍受ユーザーに対しては生成されません。

---

## ユーザーアカウントの要件

TACACS+ AAA サービスを設定する前に、次の TACACS+ サーバーと StarOS ユーザーアカウントのプロビジョニング要件に注意してください。

### TACACS+ ユーザーアカウントの要件

TACACS+ サーバーは、次の TACACS+ ユーザーアカウント情報を使用してプロビジョニングする必要があります。

- 既知の管理ユーザーのリスト。
- 各ユーザーのプレーンテキストまたは暗号化されたパスワード。
- 各ユーザーが属するグループの名前。
- ユーザーグループのリスト。
- 各グループで許可/拒否される TACACS+ の特権レベルとコマンド。



---

**重要** TACACS+ の権限レベルは、ネットワークの TACACS+ サーバーデータベースに属性値ペア (AVP) として保存されます。ユーザーは、権限レベルに関連付けられた一連のコマンドに制限されます。次の表で、TACACS+ の権限レベルを StarOS CLI の管理ロールと責任にマッピングする方法を説明します。

---

CLI 管理ロールへの TACACS+ 権限レベルのデフォルトマッピングを表示するには、Exec モードの **show tacacs priv-lvl** コマンドを実行します。デフォルトのマッピングは、StarOS リリースとビルドタイプによって異なります。

Tacacs+ の権限レベルをデフォルトの StarOS 認証値から再設定するには、TACACS+ コンフィギュレーションモードの **priv-lvl** コマンドと **user-id** コマンドを使用します。詳細については、『*Command Line Interface Reference*』の「*TACACS+ Configuration Mode Commands*」の章を参照してください。



---

**重要** FTP はサポートされていません。

---

### StarOS ユーザーアカウントの要件

システムへの管理アクセスが許可されている TACACS+ ユーザーには、StarOS で次のユーザーアカウント情報が定義されている必要があります。

- username
- password
- 管理ロールと権限





**重要** システムでユーザーと管理者の権限を定義する手順については、「システム管理ユーザーの設定」を参照してください。

## TACACS+ AAA サービスの設定

この項では、システム上の管理ユーザーに TACACS+ AAA サービスを設定する方法について説明します。



**注意** TACACS+ AAA サービスを初めて設定する場合は、管理者ユーザーが StarOS にログインするために TACACS+ 以外のサービスを使用する必要があります。そうしないと、TACACS+ ユーザーはシステムへのアクセスが拒否されます。

TACACS+ 以外のサービスを使用してシステムにログインします。

システムで TACACS+ AAA サービスを設定するには、次の例を使用します。

```
configure
tacacs mode
  server priority priority_number ip-address tacacs+srvr_ip_address
end
```

(注)

- **server priority priority\_number** : TACACS+ サーバーが TACACS+ 認証を試行する順序を指定する 1 ~ 4 の整数である必要があります。優先順位は 1 が最も高く、4 が最も低くなります。優先順位の数値は、設定されている TACACS+ サーバーに対応します。
- **ip-address** : 有効な TACACS+ サーバーの IPv4 アドレスである必要があります。これを使用し、TACACS+ AAA サービスを介してこのシステムにアクセスする管理ユーザーを認証します。
- デフォルトでは、TACACS+ 設定によって、認証、許可、およびアカウントिंगの各サービスが提供されます。

StarOS で TACACS+ を有効にします。

```
configure
aaa tacacs+
end
```

詳細については、[コンソールの TACACS+ 認証の無効化 \(91 ページ\)](#) を参照してください。

「設定の確認と保存」の章の説明に従って、設定を保存します。



**重要** すべての TACACS+ コンフィギュレーション モードのコマンドとオプションの詳細については、『*Command Line Reference*』の「*TACACS Configuration Mode Commands*」の章を参照してください。

## 非ローカル VPN 認証のための TACACS+ の設定

デフォルトでは、TACACS+ 認証はローカルコンテキストへのログインに関連付けられています。TACACS+ 認証は、非ローカルコンテキスト VPN ログイン用に設定することもできます。TACACS+ は、以下で説明するオプションを使用して設定し、有効にする必要があります。

**stop** キーワードオプションは、TACACS+ コンフィギュレーションモードの **on-unknown-user** コマンドで使用できます。TACACS+ がコマンドキーワードオプションで有効になっている場合、ユーザーがログインを試行する VPN コンテキスト名は、ユーザー名文字列で指定された VPN 名と一致している必要があります。コンテキスト名が一致しない場合、ログインは失敗して終了します。

このオプションを指定しないと、ログインシーケンスは代替ログイン方式を介して別のコンテキストで認証を試行します。たとえば、**on-unknown-user stop** の設定がない場合、管理者アカウントは、非ローカルの VPN コンテキストを介してローカルコンテキストにログインできます。ただし、**on-unknown-user stop** の設定では、ローカルコンテキストのログインは試行されず、管理者アカウントのログイン認証は失敗します。

```
configure
tacacs mode
    on-unknown-user stop &quest;
end
```

## TACACS+ の設定の確認

この項では、TACACS+ 設定の確認方法について説明します。

システム CLI からログアウトし、TACACS+ サービスを使用して再度ログインします。



**重要** StarOS で TACACS+ AAA サービスを設定して有効にすると、システムは最初に TACACS+ AAA サービスを介して管理ユーザーを認証しようとします。デフォルトでは、TACACS+ 認証が失敗した場合、システムは TACACS+ AAA 以外のサービスを使用して認証を続行します。

Exec モードのプロンプトで、次のコマンドを入力します。

```
show tacacs [ client | priv-lvl | session | summary ]
```

**show tacacs** コマンドの出力には、ユーザー名、ログイン時刻、ログインステータス、現在のセッション状態、権限レベルなどのサマリー情報がアクティブな TACACS+ セッションごとに表示されます。オプションのフィルタキーワードによって、追加情報が表示されます。

次に、このコマンドの出力例を示します。この例では、*asradmin* という名前のシステム管理ユーザーが、TACACS+ AAA サービスを介してシステムに正常にログインしています。

```
active session #1:
login username      : asradmin
login tty           : /dev/pts/1
time of login       : Fri Oct 22 13:19:11 2011
login server priority : 1
current login status : pass
```

```
current session state      : user login complete
current privilege level   : 15
remote client application : ssh
remote client ip address  : 111.11.11.11
last server reply status  : -1
total TACACS+ sessions   : 1
```



**重要** すべての TACACS+ メンテナンスコマンドの詳細については、『*Command Line Interface Reference*』を参照してください。

## TACACS+ サーバーでの IPv6 アドレスサポート

### 認証方式の分離

コンソールポートにアクセスし、SSH/telnetセッション（vty回線）を確立するための個別の認証方式を設定できます。

TACACS+ をグローバルに設定すると、コンソールおよび vty 回線へのアクセスは両方ともその方式を使用して認証されます。

コンソールポートは StarOS への最後のリゾートアクセスであるため、コンソールのローカル認証を設定し、vty 回線に TACACS+ を使用することができます。



**重要** この機能は、AAA（認証、許可、およびアカウントティング）サービス、およびローカルユーザーに対して拡張されます。たとえば、ローカルユーザーは、vty 回線を介してのみアクセス権を持つコンソールアクセスおよび AAA（VPN コンテキスト）ユーザーのみを持つことができます。

認証方式（コンソールと vty 回線）を分離するには、認証のタイプに基づいてユーザーのコンソールアクセスを無効にする必要があります。

### コンソールの TACACS+ 認証の無効化

グローバルコンフィギュレーションモードの `aaa tacacs+` コマンドの `noconsole` キーワードは、コンソール回線での TACACS+ 認証を無効にします。

```
configure
aaa tacacs+ noconsole
exit
```

デフォルトでは、TACACS+ サーバー認証は、コンソールまたは vty 回線からのログインに対して実行されます。`noconsole` を有効にすると、コンソール回線のローカルデータベース認証

を優先して TACACS+ 認証がバイパスされます。vty 回線では、TACACS+ は有効のままになります。



**重要** `aaa tacacs+ noconsole` が設定されている場合は、有効なログイン情報を持つローカルユーザーがコンソールポートにログインできます。`on-authen-fail stop` と `on-unknown-user stop` が TACACS+ コンフィギュレーションモードでイネーブルになっている場合でも、コンソールポートにログイン可能です。ユーザーが TACACS+ ユーザーでない場合、vty 回線でログインできません。

## コンソールの AAA ベースの認証を無効化

グローバル コンフィギュレーション モードの `local-user allow-aaa-authentication noconsole` コマンドの `noconsole` キーワードは、コンソール回線での AAA ベースの認証を無効にします。

### configure

```
local-user allow-aaa-authentication noconsole
exit
```

ローカルユーザー認証は、AAA ベースの認証と `local-user allow-aaa-authentication noconsole` が有効になる前に常に実行されるため、`no local-user allow-aaa-authentication` が設定されている場合と同じに動作します。VTY 回線には影響しません。



**重要** このコマンドは、ローカルで使用されているデータベースが使用できないため、信頼できるビルドには適用されません。

## コンテキストレベルで TACACS+ 認証を無効化

グローバル コンフィギュレーション モードで `aaa tacacs+` を有効にすると、TACACS+ 認証はすべてのコンテキスト（ローカルおよび非ローカル）に自動的に適用されます。一部のネットワーク展開では、特定のコンテキストに対して TACACS+ サービスを無効にすることができます。

コンテキスト内で TACACS+ サービスを無効にするには、`no aaa tacacs+` コンテキスト設定コマンドを使用します。

### configure

```
context ctx_name
no aaa tacacs+
```

以前に無効にされていたコンテキスト内で TACACS+ サービスを有効にするには、`aaa tacacs+` コンテキスト設定コマンドを使用します。



**重要** コンテキストレベルでサービスを選択的に無効にするには、その前に、グローバル コンフィギュレーションモード (すべてのコンテキスト) で AAA TACACS+ サービスを有効にする必要があります。グローバルに有効になっていない場合は、コンテキストレベルで TACACS+ サービスを選択的に有効にすることはできません。

## コンソール/VTY 回線でのローカルユーザーログインの制限

StarOS ユーザーを作成するときのセキュリティ管理者は、そのユーザーがコンソールを使用してログインするか、または VTY 回線を使用してログインするかを指定できます。グローバル コンフィギュレーションモードの `local-user username` コマンドの `[ noconsole | novty ]` キーワードはこれらのオプションをサポートしています。

### configure

```
local-user username <username> [ noconsole | novty ]
exit
```

**noconsole** キーワードを指定すると、ユーザーはコンソールポートにログインできなくなります。**novty** キーワードを指定すると、ユーザーは SSH または telnet セッションを介してログインできなくなります。どちらのキーワードも指定されていない場合は、コンソールと VTY 回線の両方へのアクセスが許可されます。



**重要** **noconsole** キーワードまたは **novty** キーワードの使用は、新しいローカルユーザーデータベース形式でのみサポートされています。**update local-user database** を実行していない場合は、これらのキーワードを有効にする前に実行する必要があります。そうしないと、**noconsole** キーワードと **novty** キーワードはローカルユーザーデータベースに保存されません。システムの再起動後も、すべてのユーザーがコンソールと VTY 回線にアクセスできます。詳細については、[ローカルユーザーデータベースの更新とダウングレード \(84 ページ\)](#) を参照してください。



**重要** このコマンドは、ローカルで使用されているデータベースが使用できないため、信頼できるビルドには適用されません。

## AAA ベースのユーザーのコンソールアクセスの制限

AAA ベースのユーザーは、通常、VTY 回線を介してログインします。ただし、一部のユーザーはコンソール回線にだけアクセスするように制限することもできます。ローカルユーザーデータベースを使用しない場合 (または信頼できるビルドを実行している場合) は、他の AAA ベースのユーザーについてはコンソール回線へのアクセスを制限することでこれを行う必要があります。コンソール回線にアクセスできないすべてのレベルの管理者ユーザーに対して **noconsole** キーワードを有効にします。

**noconsole** キーワードは、次に示すコンテキスト コンフィギュレーションモード コマンドで使用できます。

#### configure

```
context <ctx_name>
  administrator <username> { encrypted | nopassword | password } noconsole

  config-administrator <username> { encrypted | nopassword | password }
noconsole
  inspector <username> { encrypted | nopassword | password } noconsole
  operator <username> { encrypted | nopassword | password } noconsole
exit
```

**noconsole** キーワードを指定すると、コンソール回線へのユーザーアクセスが無効になります。デフォルトでは、**noconsole** は有効になっていないため、すべての AAA ベースのユーザーがコンソール回線にアクセスできます。




---

**重要** **local-user allow-aaa-authentication noconsole** コマンドが優先されます。この場合、すべての AAA ベースのユーザーがコンソール回線にアクセスできません。

---

## 設定変更の確認

Exec モードの **show configuration** コマンドを使用して、認証方式の分離に関連した変更を確認できます。設定変更を保存した後、**show configuration | grep noconsole** および **show configuration | grep novty** を実行します。これらのコマンドの出力には、実行したすべての変更が示されます。

## シャーシキーの設定

シャーシキーは、システムごとに設定する必要があります。このキーは、コンフィギュレーション ファイルにある暗号化されたパスワードを復号するために使用されます。

### 概要

シャーシキーは、設定ファイル内の暗号化されたパスワードを暗号化および復号するために使用されます。2 つ以上のシャーシが同じシャーシキー値で設定されている場合、暗号化されたパスワードは、同じシャーシキー値を共有するシャーシのいずれかで復号できます。そのため、特定のシャーシキー値は、異なるシャーシキー値で暗号化されたパスワードは復号できません。

シャーシキーは、ファイルに保存され、設定ファイル内の機密データ（パスワードや秘密など）を保護するためのマスターキーとして使用されるシャーシ ID を生成するために使用されます。

シャーシ ID はシャーシキーの SHA256 ハッシュです。シャーシキーは、CLI コマンドまたはクイックセットアップウィザードを使用してユーザーが設定できます。シャーシ ID が存在しない場合、シャーシ ID を生成するにはローカル MAC アドレスを使用します。

ユーザーはクイックセットアップウィザードまたは CLI コマンドを使用して、シャーシキーを明示的に設定する必要があります。設定されていない場合、ローカル MAC アドレスを使用してもデフォルトのシャーシ ID は生成されません。シャーシキー（つまり、シャーシ ID）がない場合、機密データは保存された設定ファイルには表示されません。シャーシ ID は、ユーザーが入力したシャーシキーおよび 32 バイトのセキュアな乱数による SHA256 ハッシュ（base36 形式で暗号化）です。これにより、キーセキュリティを得るためにシャーシキーとシャーシ ID には 32 バイトのエントロピーが確保されます。

シャーシ ID を使用できない場合、設定ファイル内の機密データの暗号化と復号は機能しません。

## 新しいシャーシキーの値の設定

### CLI コマンド



---

**重要** **chassis key value** コマンドと **chassis keycheck** コマンドが実行できるのは、セキュリティ管理者権限を持つユーザーのみです。

---

新しいシャーシキーを入力するには、Exec モードの **chassis key value** *key\_string* コマンドを使用します。

*key\_string* は、1 ～ 16 文字の英数字文字列です。シャーシキーは、パスワードのように一方向の暗号化された値として保存されます。このため、シャーシキーの値がプレーンテキスト形式で表示されることはありません。

Exec モードの **chassis keycheck** *key\_string* コマンドは、入力した *key\_string* に基づいて一方向の暗号化されたキー値を生成します。生成された暗号化キーの値は、以前に入力したシャーシキー値の暗号化されたキー値と比較されます。暗号化された値が一致すると、コマンドは成功し、キーチェックに合格します。比較に失敗した場合は、キーチェックが失敗したことを示すメッセージが表示されます。デフォルトのシャーシキー（MAC アドレス）が現在使用されている場合、比較するシャーシキー値がないため、このキーチェックは常に失敗します。

複数のシャーシが同じシャーシキー値を共有しているかどうかを確認するには、**chassis keycheck** コマンドを使用します。



---

**重要** 既存のシャーシ ID ファイルが存在しない場合、**chassis keycheck** コマンドは非表示になります。

---

詳細については、『*Command Line Interface Reference*』の「*Exec Mode Commands*」の章を参照してください。

シャーシ ID は、より安全なアルゴリズムを使用して、シャーシキーから生成されます。結果として得られる 44 文字のシャーシ ID は、同じファイルに保存されます。

シャーシ ID ファイルがすでに存在しているシャーシでは、何も変更されません。ただし、両方の管理カードにシャーシ ID ファイルがなくなった場合、既存のすべての設定ファイルが無効になります。元の値と同じ新しいシャーシキーを入力しても、シャーシ ID の生成には新しい方法が使用されているため、問題は解決しません。



**注意** 新しいシャーシキーを設定した後、リロードを開始する前に設定を保存する必要があります。「設定の確認と保存」の章を参照してください。

## クイックセットアップウィザード

シャーシ ID ファイルが存在しない場合、クイックセットアップウィザードにより、ユーザーにシャーシキーを入力するように求めるプロンプトが表示されます。シャーシキーが入力されていない場合、デフォルトのシャーシ ID は生成されません。

クイックセットアップウィザードを実行するには、Exec モードの **setup** コマンドを実行します。

```
[local]host_name# setup
1. Do you wish to continue with the Quick Setup Wizard[yes/no]: Y
2. Enable basic configuration[yes/no]: Y
3. Change chassis key value[yes/no]: Y
4. New chassis key value: key_string
```

## FSC ファブリックの自動リセットの有効化

デフォルトでは、スイッチファブリック内で過剰な数の破棄されたファブリック出力パケットが発生した場合、ファブリック ストレージカードを手動でリセットしてファブリックをリカバリする必要があります。

過剰な数の破棄されたファブリック出力パケットが検出された場合に、必要に応じて FSC の自動リセットを有効にすることができます。

グローバル コンフィギュレーション モードの **fabric fsc-auto-recover** コマンドは、過剰な数の破棄されたファブリック出力パケットを検出したときに、自動 FSC リセットを有効または無効にします。

次のコマンドシーケンスは、この機能を有効にします。

```
configure
  fabric fsc-auto-recovery { disable | enable } [ max-attempts [
number_attempts | unlimited ] ]
end
```



**max-attempts** [ *number\_attempts* | **unlimited** ] は、StarOS が各 FSC のリセットを試行する回数を 1～99 の整数または無制限で指定します（FSC がリセットされるまで停止しません）。デフォルトの設定は 1 秒です。



---

**重要** この機能を有効にするには、最初にグローバルコンフィギュレーションモードの **fabric egress drop-threshold** コマンドを介して、ファブリック出力ドロップしきい値を設定する必要があります。

---





## 第 5 章

# モードロックメカニズムの設定

この章では、StarOS コンフィギュレーション モード内で管理ロックのメカニズムがどのように動作するかについて説明します。

ここで説明する内容は、次のとおりです。

- [モードロック設定の概要 \(99 ページ\)](#)
- [Exclusive-Lock の要求 \(100 ページ\)](#)
- [URL スクリプトに対するロック設定の効果 \(102 ページ\)](#)
- [設定ファイルの保存 \(102 ページ\)](#)
- [リロードコマンドとシャットダウンコマンド \(103 ページ\)](#)
- [show administrators コマンド \(104 ページ\)](#)

## モードロック設定の概要

Exec モードの **configure** コマンドを介して、グローバルコンフィギュレーション (**config**) モードを開始します。デフォルトでは、すべての管理ユーザーがコンフィギュレーションモードを共有します。複数の管理ユーザーは、同時にコンフィギュレーションモードへのアクセスを共有できます。これは **shared-lock** と呼ばれます。

**shared-lock** の存在を示すプライマリ表示は、コンフィギュレーションモードを開始すると表示されるメッセージです。

```
Warning: One or more other administrators may be configuring this system
```



(注) **shared-lock** の下でコンフィギュレーションモードに入った場合、デフォルトの制限的な動作変更はありません。

複数の管理者が実行設定を編集または保存すると、同時変更によって、競合、不整合、または欠落している設定コマンドが発生する可能性があります。システムの再起動を試行している場合は、設定を保存するときに同様の問題が発生する可能性があります。

**configure** コマンドのオプションの **ock [ force | warn ]** キーワードを使用すると、管理者はコンフィギュレーションモードの相互排他ロックを要求して、他のユーザーが同時に設定を変更し

ないようにすることができます。これは `exclusive-lock` と呼ばれます。管理者に `exclusive-lock` が許可されると、その他のユーザーはロックがかかっている間、セッションの期間にコンフィギュレーションモードにアクセスできません。`exclusive-lock` は、ロックを保持しているユーザーが `Exec` モードに戻った場合にのみ終了します。

設定の保存中に他のユーザーがシステムをリロードまたはシャットダウンしないようにするために、設定の保存操作中に `shutdown-lock` が有効になります。

`shared-lock`、`shared-lock`、および `shutdown-lock` などのコンフィギュレーションモードのロックメカニズムによって、競合するコマンド、ファイルの破損、およびリブートの問題が軽減されます。

## Exclusive-Lock の要求



**重要** 管理者が排他的ロックを保持してコンフィギュレーションモードを終了できないことに起因する混乱を回避するには、CLIセッション絶対タイムアウトやアイドルタイムアウトを使用してすべての管理者アカウントを設定することがベストプラクティスですこれらのタイムアウトの設定の詳細については、このガイドの「使用する前に」の章の「*CLI*を使用した初期設定」の項を参照してください。

`Exec` モードの `configure lock` コマンドを実行することにより、コンフィギュレーションモードで排他的ロックを要求できます。

```
[local]host_name# configure [ <url> ] lock [ force | warn ]
```

URL を指定すると、排他的ロックはプリロードされたコンフィギュレーションファイルに関連付けられます。URL を指定しない場合は、実行コンフィギュレーションに排他的ロックが許可されます。詳細については、[URL スクリプトに対するロック設定の効果 \(102 ページ\)](#) を参照してください。

**force** オプションを適用すると、他のすべての管理者は、現在排他的ロックを保持しているユーザーを含め、コンフィギュレーションモードを終了します。

**warn** オプションは、他のすべての管理者に対して、コンフィギュレーションモードを終了するよう警告します。この管理者は、すぐに排他的ロックを取得します。このオプションは、実際に管理者がコンフィギュレーションモードを強制的に終了する前に使用することができます。

コンフィギュレーションモードで他の管理者がいない場合は、`configure lock` を入力すると、ただちに排他的ロックが許可されます。

```
[local]host_name# configure lock
Info: No one else can access config mode while you have the exclusive lock
[local]host_name#
```

排他的ロックが許可されている場合、他の管理者はコンフィギュレーションモードへの入力や、コンフィギュレーションファイルのロードを許可されません。他の管理者がコンフィギュ

レーションモードを開始したり、コンフィギュレーションファイルをロードしたりしようとすると、次のメッセージが表示されます。

```
Failure: User <username> has the exclusive lock
- please enter 'show administrators' for more information
```

すでに排他的ロックが有効になっている場合、他の管理者が排他的ロックでコンフィギュレーションモードを開始しようとすると、次のメッセージが表示されます。

```
Failure: Another administrator is still in configuration mode
- please enter 'show administrators' for more information
```

最初に排他的ロックを取得しない場合、**configure lock force** を使用できます。

**configure lock force** が成功した場合、終了して Exec モードへ戻ることを強制されたすべてのユーザーには、コンフィギュレーションモードから強制終了されたことを示す次の警告メッセージが表示されます。

```
[local]host_name(config)#
Warning: Administrator <username> has forced you to exit from configuration mode
[local]host_name#
```

他の管理者がパスワードの入力中であつたり、中断できない重要なシステムオペレーションを実行したりする可能性が非常に低いため、**configure lock force** コマンドは成功しない場合があります。この場合、エラーメッセージが表示されます。

```
[local]host_name# configure lock force
Failure: Another administrator could not release the configuration mode lock
- please enter 'show administrators' for more information
```

**configure lock warn** コマンドは、すべてのコンフィギュレーションモードのユーザー（存在する場合）に警告メッセージを送信し、その後、最大 10 秒間待機して排他的ロックを取得します。まだコンフィギュレーションモードのユーザーがいる場合、コンフィギュレーションモードは共有ロック状態のままになります。

```
[local]host_name# configure lock warn
please wait for this message to be sent to the other administrators.....
[local]host_name(config)#
```

他の管理者のセッション出力には、最終的に次のメッセージが表示されます。

```
[local]host_name(config)#
Administrator <username> requires exclusive access to configuration mode
>>> You need to exit from configuration mode as soon as possible <<<
[local]host_name#
```

他の管理者が通常、セッション出力内のメッセージを確認することを想定していないため、**configure lock warn** コマンドを実行しても、通常は排他的ロックが取得されることはありません。



**重要** StarOS は、すべての主要なコンフィギュレーションモードのロックの連携動作をイベントログと syslog ファシリティ（設定されている場合）に記録します。発生した連携動作の記録にいつでもアクセスできます。

## URL スクリプトに対するロック設定の効果

**configure** <url> コマンドを使用して設定スクリプトファイルをロードしようとする場合は、共有ロック（デフォルト）または排他ロックのいずれかを取得する必要があります。設定スクリプトファイルには通常、**config** コマンドが含まれているため、実際には **config** コマンドが解析されて実行される前と後にロックは保留されます。

ロックは、設定ファイル全体の実行中は保持されます。双方向設定モードロックと同じ共有ロックが使用されるため、警告メッセージが表示され、（**-noconfirm** が有効になっていない場合は）その後確認プロンプトが表示されます。次に例を示します。

```
[local]host_name# config /flash/myconfig.cfg
Warning: One or more other administrators may be configuring this system
Are you sure? [Y/N]:
```

**-noconfirm** が有効になっている場合は、すべてのコマンドも画面にエコーされるため、警告メッセージが画面から消え、気づかない場合もあります。



---

**重要** StarOS が最初に起動すると、初期起動設定ファイルはロード中は常に排他的にロックされません。

---

## 設定ファイルの保存

部分的または不完全なコンフィギュレーションファイルを保存すると、保存された設定が後でロードされたときに StarOS が不安定になる可能性があります。StarOS は、変更処理中の設定を保存することをユーザーに禁止します。

保存操作中に共有ロックが有効になっている場合、保存操作の確認を求めるプロンプトが表示されます。

```
[local]host_name# save configuration /flash/config.cfg
Warning: One or more other administrators may be configuring this system
Are you sure? [Y/N]:
```

排他的ロックがユーザーによって保持されている場合、保存操作は失敗します。

```
[local]host_name# save configuration /flash/config.cfg
Failure: Configuration mode is currently locked, use ignore-lock to ignore lock
```

既存の排他ロックを上書きするには、**save configuration** コマンドで **ignore-lock** キーワードを使用します。

```
[local]host_name# save configuration /flash/config.cfg ignore-locks
Warning: Ignoring the configuration mode lock held by another administrator
```



**重要** また、**save configuration** コマンドを使用すると、設定の保存中に他のユーザーがシステムをリロードまたはシャットダウンできなくなるシャットダウンロックも有効になります。詳細については、「[リロードコマンドとシャットダウンコマンド \(103ページ\)](#)」を参照してください。

## リロードコマンドとシャットダウンコマンド

Exec モードの **reload** コマンドと **shutdown** コマンドについては、**save configuration** コマンドの進行中にこれらのいずれかのコマンドを実行すると、設定ファイルが破損するか、または不完全なものになる可能性があります。

この問題の発生を防ぐため、**reload** コマンドと **shutdown** コマンドは、StarOS 全体ですべての **save configuration** コマンドを実行している状態で CLI shutdown-lock を共有します。つまり、何らかの **save configuration** コマンドの実行中は、StarOS は **reload** コマンドや **shutdown** コマンドを実行できません。これらのコマンドは、すべての保存設定操作が完了するまで無期限にキューに入れます。

ユーザーが待機キュー内で無期限に「ハング」するのを防ぐために、ユーザーは Ctrl+C を押して、次の例に示すように待機を終了することができます。

```
[local]host_name# reload
Are you sure? [Yes|No]: yes
Waiting for other administrators to finish saving configuration
(ctrl-c to abort) .....^C
Action aborted by ctrl-c
[local]host_name#
```

保存中のプロセスでファイルを破損するリスクがあるかどうかにかかわらず、StarOS をすぐに再起動する必要がある場合がまれにありますが、このような場合に **reload** コマンドまたは **shutdown** コマンドを **ignore-locks** キーワードと組み合わせて使用することができます。このオプションを選択すると、StarOS は適切な警告メッセージを表示しますが、再起動を開始する前に設定の保存操作が完了するまで待機しません。

```
[local]host_name# reload ignore-locks -noconfirm
Warning: One or more other administrators are saving configuration
Starting software 21.0...
Broadcast message from root (pts/2) Wed May 11 16:08:16 2016...
The system is going down for reboot NOW !!
```



**注意** システムを再起動するときに **ignore-locks** キーワードを指定すると、設定ファイルが破損する可能性があります。

## show administrators コマンド

Exec モードの **show administrators** コマンドには、管理者のセッションの現在のロックモードを示す 1 文字の「M」列があります。M モードの文字は、次のように定義されます。

- [空白]: 管理者は Exec モードを使用しています。
- **c**: 管理者セッションは現在コンフィギュレーションモードになっています（共有ロック）。
- **s**: 管理者セッションは現在、設定を保存しています。
- **f**: 管理者セッションは現在、設定ファイルをロードしています。
- **L**: 管理者セッションは現在、排他的ロックを使用してコンフィギュレーションモードになっています。

次に、現在のロックモードを示す **show administrators** コマンドの出力例を示します。

```
[local]asr5500# show administrators
Administrator/Operator Name  M Type      TTY          Start Time
-----
Bob                          admin      /dev/pts/2   Tue Mar 29 11:51:15 2016
Alice                        c admin      /dev/pts/1   Mon Mar 28 14:41:15 2016
Carol                        admin      /dev/pts/0   Mon Mar 28 14:40:52 2016
```





## 第 6 章

# 管理設定

この章では、Object Request Broker Element Management (ORBEM) および Simple Network Management Protocol (SNMP) オプションを設定する手順について説明します。

この章は、次の項で構成されています。

- [SNMP MIB ブラウザ \(105 ページ\)](#)
- [SNMP サポート \(107 ページ\)](#)

## SNMP MIB ブラウザ

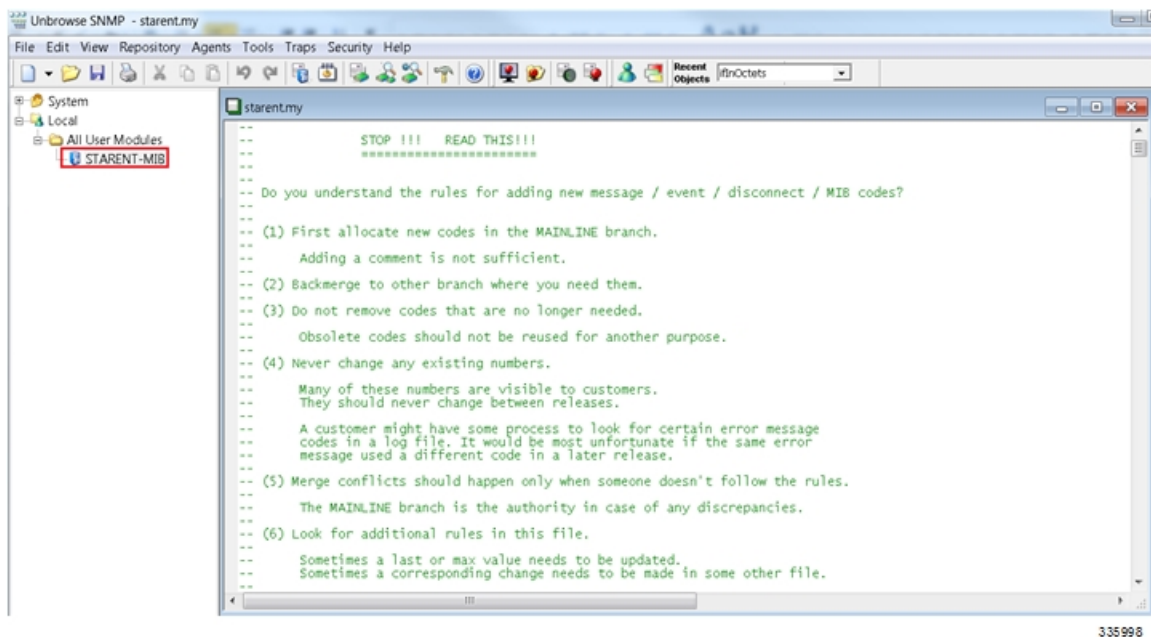
この項では、MIB ブラウザを使用して最新の Cisco Starent MIB ファイルにアクセスする手順について説明します。更新された MIB ファイルは、すべての StarOS リリースに付属しています。アカウントの設定とファイルへのアクセスについては、シスコの販売担当者またはサービス担当者にお問い合わせください。

MIB ブラウザを使用すると、SNMP 対応デバイスからデータを引き出すことができます。標準 MIB と独自 MIB をロードできます。このツールを使用すると、MIB データを読み取り可能な形式で表示したり、特定の OID を検索する機能を提供したりすることができます。ブラウザに MIB ツリー内のすべての MIB が表示されます。これにより、すべてのオブジェクト、トラップ、または適合性を簡単に見つけて識別することができます。

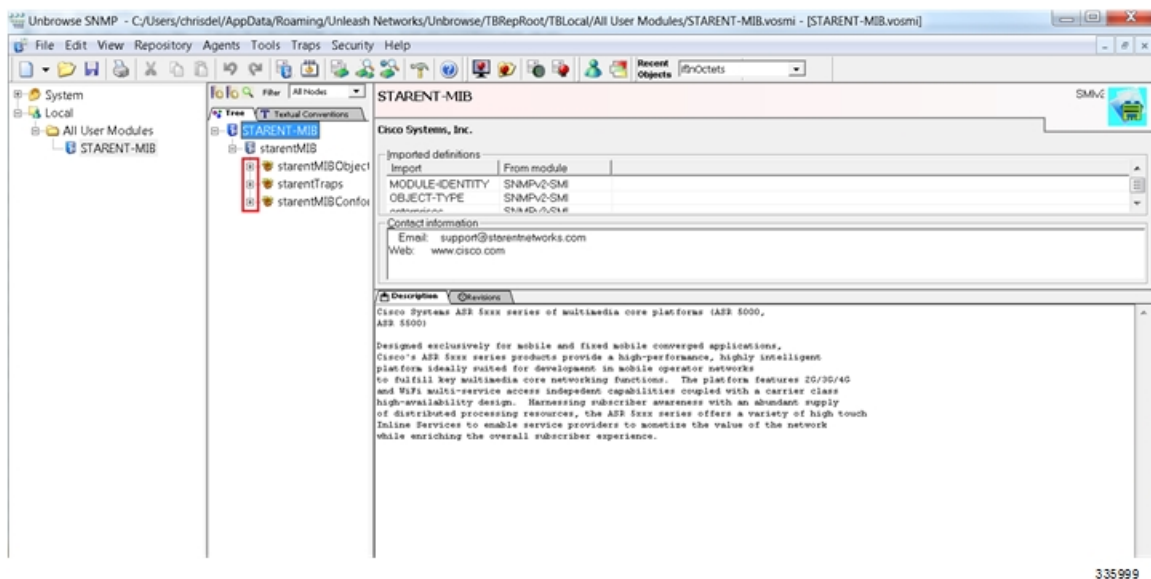
特定の StarOS ビルドの SNMP MIB を表示するには、次の手順を実行します。

- ステップ 1** 特定の StarOS リリースの MIB ファイルへのアクセス権を取得するには、シスコのセールス担当者またはサービス担当者にお問い合わせください。
- ステップ 2** 圧縮されたコンパニオンファイルをデスクトップ上のフォルダにダウンロードします。ファイル名は、次の規則に従います。 **companion\_xx.x.x.tgz**
- ステップ 3** コンパニオンファイルを開き、解凍して、同じフォルダに展開します。
- ステップ 4** **companion-xx.x.x.xxxxx** ファイルフォルダをダブルクリックします。
- ステップ 5** **companion-xx.x.x.xxxxx.tar** を解答して展開します。
- ステップ 6** MIB ブラウザで、.tar ファイル内の **starent.my** ファイルを検索して開きます。SNMP MIB ブラウザを使用すると、MIB **.my** ファイルを表示する前にコンパイルできます。

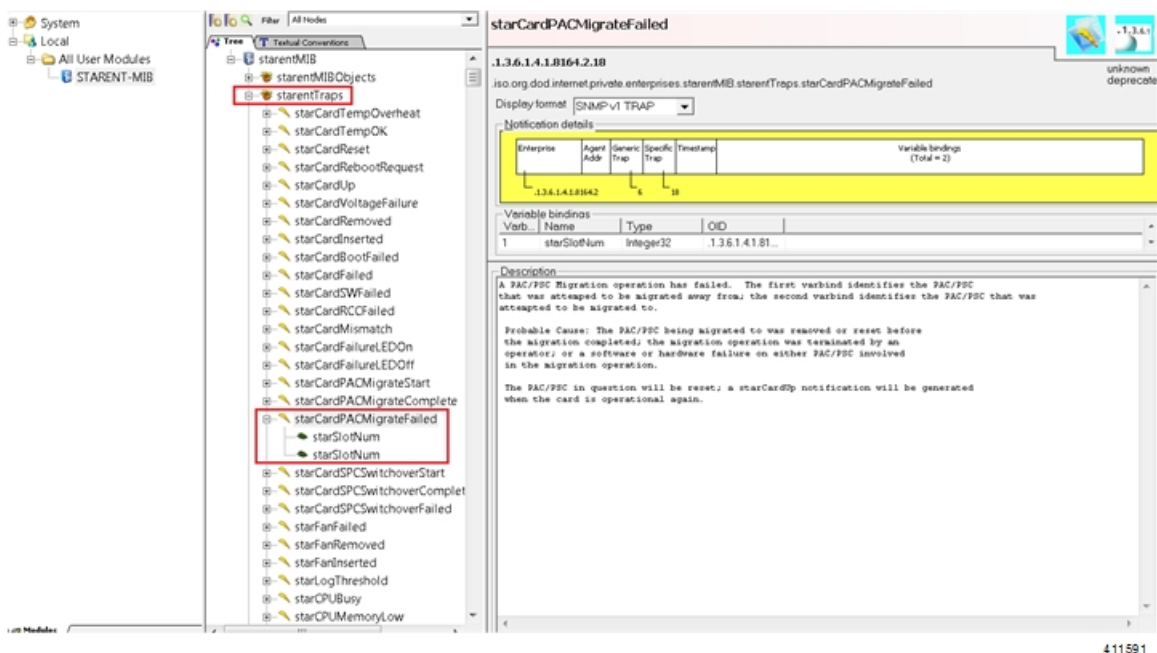
ステップ7 MIB ファイルをコンパイルするには、STARENT-MIB ファイルをクリックして、[File]>[Open] を選択します。



STARENT-MIB.vosmi ファイルが開きます。



次の例で示すMIBブラウザのツリー図には、各オブジェクト、トラップ、および適合性の詳細を表示することができます。この例には、starCardPACMigrateFailedトラップのOID番号とトラップの詳細が含まれています。



SNMP MIB ブラウザでは、特定の MIB を検索することができます。特定の OID（オブジェクト識別子）を検索して、特定の MIB エントリを検索できます。

**重要** 特定のリリースの SNMP MIB の変更については、該当するバージョンの『*Release Change Reference*』の「*SNMP MIB Changes in Release xx*」の章を参照してください。

## SNMP サポート

システムは SNMP を使用して、ネットワーク上の EMS サーバーまたはアラームサーバーにトラップまたはイベントを送信します。これらのデバイスと通信するには、SNMP を設定する必要があります。



**重要** この項の設定例で使用されているコマンドは、基本機能を提供します。最も一般的なコマンドとキーワードのオプションが示されています。多くの場合、他のオプションコマンドやキーワードオプションを使用できます。詳細については、『*Command Line Interface Reference*』を参照してください。

『*SNMP MIB Reference*』では、StarOS でサポートされている MIB と SNMP のトラップが説明されています。

システムが EMS サーバーまたはアラームサーバーと通信するように設定するには、次の手順を実行します。

- ステップ1 の設定例を適用して、UDPポートやアラームサーバーターゲットなどのSNMPパラメータを設定します。  
[SNMP とアラームサーバーパラメータの設定 \(108 ページ\)](#)
- ステップ2 新しいSNMP設定を表示するには、次の手順を実行します。[SNMPパラメータの確認 \(109 ページ\)](#)
- ステップ3 「設定の確認と保存」の説明に従って、設定を保存します。

## SNMP とアラームサーバーパラメータの設定

SNMP およびアラームサーバーパラメータを設定するには、次の例を使用します。

```
configure
system contact contact_name
system location location_name
snmp authentication-failure-trap
snmp community community_string
snmp server port port_number
snmp target name ip_address
snmp engine-id local id_string
snmp notif-threshold value low low_value period time_period
snmp user user_name
snmp mib mib_name
snmp runtime-debug [ debug-tokens token_id token_id token_id...token_id
end
```

注：

- システム管理者は、エラー状態を示すトラップが生成されたときに連絡する担当者の名前です。
- **snmp コミュニティ** 文字列は、システムの Management Information Base (MIB) へのアクセスを可能にするパスワードです。
- システムは、SNMPv1 トラップ、SNMPv2c トラップ、または SNMPv3 トラップを多数のターゲットデバイスに送信できます。ただし、EMS は SNMP バージョン 1 (SNMPv1) および SNMP バージョン 2c (SNMPv2c) トラップのみを処理できます。設定している SNMP ターゲットが EMS アプリケーションである場合は、**snmp target** コマンドを使用してバージョン 1 またはバージョン 2c の使用を設定します。複数のターゲットを設定するのに必要な回数だけこのコマンドを発行します。複数のターゲットを設定した場合、生成されたアラームは、設定されたすべてのターゲットに送信されます。
- **snmp notif-threshold** コマンドは、通知が SNMP ユーザーに伝達される前に、特定のイベントに対して生成する必要がある SNMP 通知の数 (デフォルトは 300) と、モニタリングウィンドウサイズの秒数 (デフォルトは 300) を設定します。
- **snmp engine-id local** コマンドはオプションです。これは、ネットワークで SNMP v3 サポートが必要な場合にのみ必要です。エンジン ID は、SNMP エンジンと関連する SNMP エンティティを一意に識別するため、2 つの間にセキュリティ アソシエーションを提供し、データの送受信を行います。

- **snmp ユーザー**の名前は SNMP v3 用であり、オプションです。このコマンドには、多数のキーワードオプションが関連付けられています。
- 他の業界標準および Cisco MIB を有効にするには、**snmp mib** コマンドを使用します。デフォルトでは、STARENT-MIB のみが有効になっています。
- デフォルトでは、SNMP ランタイムデバッグは常に実行され、イベントロギングのために CPU サイクルを消費します。CPU 使用率を制御するには、**no snmp runtime-debug** を設定して、ランタイムデバッグを無効にすることができます。このコマンドのオプションを使用すると、指定された MIB を参照して解析する SNMP トークン値を指定できます。



**重要** SNMPv3 トラップは、一部の EMS アプリケーションではサポートされていない場合があります。

## SNMP パラメータの確認

**ステップ 1** SNMP サーバー情報が正しく設定されていることを確認するには、**show snmp server** コマンドを実行します。次に、このコマンドの出力例を示します。

```
SNMP Server Configuration:
  Server State           : enabled
  SNMP Port             : 161
  sysLocation           : chicago
  sysContact            : admin
  authenticationFail traps : Enabled
  EngineID              : 123456789
  Alert Threshold       : 100 alerts in 300 seconds
  Alert Low Threshold   : 20 alerts in 300 seconds
SNMP Agent Mib Configuration:
  STARENT-MIB          : Enabled
  IF-MIB               : Disabled
  ENTITY-MIB           : Disabled
  ENTITY-STATE-MIB    : Disabled
  ENTITY-SENSORE-MIB  : Disabled
  HOST-RESOURCES-MIB  : Disabled
  CISCO-MOBILE-WIRELESS-SERVICE-MIB : Disabled
  CISCO-ENTITY-DISPLAY-MIB : Disabled
  CISCO-PROCESS-MIB   : Disabled
  CISCO-ENTITY-FRU-CONTROL-MIB : Disabled
```

**ステップ 2** 次のコマンドを入力して、SNMP コミュニティが正しく設定されていることを確認します。

```
show snmp communities
```

このコマンドの出力には、設定されている SNMP コミュニティとそれに対応するアクセスレベルが一覧表示されます。

**ステップ 3** 次のコマンドを入力して、SNMP トランスポートが正しく設定されていることを確認します。

```
show snmp transports
```

次に、出力例を示します。

```
Target Name:   rms1
IP Address:    192.168.1.200
Port:          162
Default:       Default
Security Name: public
Version:       1
Security:
View:
Notif Type:    traps
```

## SNMP トラップ生成の制御

システムは、SNMP トラップ（通知）を使用して、特定のイベントが発生したことを示します。デフォルトでは、システムですべてのトラップの生成が可能です。ただし、個々のトラップを無効にして、特定のタイプのトラップやアラームレベルのトラップだけを生成することもできます。この項では、SNMP トラップを無効または有効にする手順について説明します。



**重要** この項の設定例で使用されているコマンドは、基本機能を提供します。最も一般的なコマンドとキーワードのオプションが示されています。多くの場合、他のオプションコマンドやキーワードオプションを使用できます。すべてのコマンドの詳細については、『*Command Line Interface Reference*』を参照してください。

SNMP トラップの生成を設定するには、次の手順を実行します。

**ステップ 1** 次の設定例を適用して、パラメータを設定します。

```
configure
  snmp trap suppress
  snmp trap suppress trap_name1 trap_name2 ... trap_nameN
```

抑制していたトラップを後で再度有効にする場合は、**snmp trap enable** コマンドを使用します。

**ステップ 2** 「設定の確認と保存」の説明に従って、設定を保存します。



## 第 7 章

# 設定の確認と保存

この章では、システム設定の保存方法について説明します。

- [設定の確認 \(111 ページ\)](#)
- [ファイルシステムの同期 \(113 ページ\)](#)
- [設定の保存 \(113 ページ\)](#)

## 設定の確認

いくつかのコマンドを使用して、機能、サービス、またはシステムの設定を確認できます。多くはその実装において階層型であり、コンフィギュレーションファイル内の一部または特定の行に固有のものもあります。

## 機能設定

多くの構成では、特定の機能を設定して確認する必要があります。例には、IP アドレスプールの設定が含まれています。次の例を使用して、リストされているコマンドを入力し、機能設定が適切であることを確認します。

IP アドレスプールの設定を表示するには、**show ip pool** コマンドを入力します。このコマンドの出力は、次に示すサンプルのようになります。この例では、すべての IP プールが *isp1* コンテキストで設定されています。

```
context : isp1:
+-----Type:          (P) - Public          (R) - Private
|                     (S) - Static          (E) - Resource
|
|+-----State:       (G) - Good            (D) - Pending Delete      (R)-Resizing
||
||+---Priority:       0..10 (Highest      (0) .. Lowest (10))
||||
||||+---Busyout:     (B) - Busyout configured
|||||
vvvvvv Pool Name          Start Address  Mask/End Address  Used      Avail
-----
PG00 ipsec                12.12.12.0     255.255.255.0    0         254
PG00 pool1              10.10.0.0      255.255.0.0     0         65534
SG00 vpnpool            192.168.1.250  92.168.1.254    0         5
```

Total Pool Count: 5



**重要** システムの機能を設定するには、これらの機能専用の **show** コマンドを使用します。詳細については、『*Command Line Interface Reference*』の「*Exec Mode show Commands*」の章を参照してください。

## サービス構成

次のコマンドを入力して、サービスの作成と適切な設定がされていることを確認します。

```
show service_type service_name
```

出力は、次に示す例のように、サービスパラメータの設定を簡潔にリストしたものです。この例では、*pgw* という名前の P-GW サービスが設定されています。

```
Service name           : pgw1
Service-Id             : 1
Context                : test1
Status                 : STARTED
Restart Counter        : 8
EGTP Service           : egtpl
LMA Service            : Not defined
Session-Delete-Delay Timer : Enabled
Session-Delete-Delay timeout : 10000(msecs)
PLMN ID List           : MCC: 100, MNC: 99
Newcall Policy         : None
```

## コンテキストの設定

**show context name name** コマンドを入力して、コンテキストが作成されており、正しく設定されていることを確認します。

出力にはアクティブなコンテキストが表示されます。ID は次に示す例と同様になります。この例では、*test1* というコンテキストが設定されています。

```
Context Name      ContextID      State
-----
test1             2                Active
```

## システム設定 (System Configuration)

**show configuration** コマンドを入力して、設定ファイル全体が作成され、正しく設定されていることを確認します。

このコマンドは、上記で定義したコンテキストとサービス設定を含む設定全体を表示します。



## 設定エラーの検出

コンフィギュレーション ファイルのエラーを特定するには、**show configuration errors** コマンドを入力します。

このコマンドは、設定内でされた検出されたエラーを表示します。たとえば、「service1」という名前のサービスを作成していて、設定の別の部分で「srv1」と入力すると、このエラーが表示されます。

設定の特定のセクションを指定するには、このコマンドを調整する必要があります。次の例に示すように、**section** キーワードを追加し、[help] メニューからセクションを選択します。

```
show configuration errors section ggsn-service
```

または

```
show configuration errors section aaa-config
```

設定にエラーが含まれていない場合は、次のような出力が表示されます。

```
#####  
Displaying Global  
AAA-configuration errors  
#####  
Total 0 error(s) in this section !
```

## ファイルシステムの同期

次の Exec モードコマンドを入力して、ローカルファイルシステムを同期します。

```
[local]host_name# filesystem synchronize all
```

**filesystem** コマンドは複数のキーワードをサポートしており、ファイルシステムの破損を確認および修復したり、ファイルシステムを特定のストレージデバイスと同期したりできます。詳細については、『*Command Line Interface Reference*』の「*Exec Mode Commands*」の章をご覧ください。

## 設定の保存

次の手順では、Exec モードのルートプロンプトが表示されていることを前提としています。

```
[local]host_name#
```

現在の設定を保存するには、次のコマンドを入力します。

```
save configuration url [ obsolete-encryption | showsecrets | verbose ]  
[ -redundant ] [ -noconfirm ]
```

*url* は、コンフィギュレーション ファイルを保存する場所を指定します。ローカルファイルまたはリモートファイルを参照する場合があります。



---

**重要** 次の URL フィールド（ディレクトリ、ファイル名、ユーザー名、パスワード、ホスト、またはポート番号）に文字列を入力する場合は、「/」（スラッシュ）、「:」（コロン）、または「@」（アットマーク）の文字を使用しないでください。

---



---

**重要** **obsolete-encryption** キーワードおよび **showsecrets** キーワードは、StarOS 19.2 以降の **save configuration** コマンドから削除されました。削除されたキーワードを含むスクリプトまたは設定を実行すると、警告メッセージが生成されます。

---



---

(注) このコマンドでは **usb1** キーワードオプションおよび **usb2** キーワードオプションを使用できますが、このオプションは、デバイスがハイパーバイザを介してサーバーに設定されている場合にのみ使用できます。これには、仮想コントローラの作成と使用可能なデバイスの指定が含まれます。

推奨手順は、VPC 設定を外部ネットワークデバイスに保存することです。

---

上記のコマンドの詳細については、『*Command Line Interface Reference*』の「*Exec Mode Commands*」の章を参照してください。

```
save configuration /flash/cfgfiles/system.cfg
```



## 第 8 章

# システムインターフェイスとポート

この章では、コンテキストを作成し、コンテキスト内でシステムインターフェイスとポートを設定する方法について説明します。これらの手順を開始する前に、製品の設定情報について、製品固有のアドミニストレーションガイドを参照してください。

- [コンテキスト \(115 ページ\)](#)
- [イーサネットインターフェイスとポート \(116 ページ\)](#)
- [VLANs \(119 ページ\)](#)

## コンテキスト

特定の機能を実行するように複数のコンテキストを設定できる場合でも、それらはすべて同じ手順を使用して作成されます。

## コンテキストの作成



**重要** この項の設定例で使用されているコマンドは、最もよく使用されているコマンドまたはその可能性の高いコマンド、および/またはキーワードオプションです。多くの場合、他のコマンドやキーワードオプションを使用できます。すべてのコマンドの詳細については、『*Command Line Interface Reference*』を参照してください。

コンテキストを作成するには、次の設定例を適用します。

```
configure  
  context name  
  end
```

追加のコンテキストを設定するには、この手順を繰り返します。

## コンテキストの表示と確認

**ステップ1** 次のコマンドを入力して、コンテキストが正常に作成されたことを確認します。

```
[local]host_name# show context all
```

出力は、次の例のような2列のテーブルです。次の例は、2つのコンテキストが作成されたことを示しています。1つは「送信元」、もう1つは「接続先」という名前です。

Context Name	ContextID	State
local	1	Active
source	2	Active
destination	3	Active

左側の列には、現在設定されているコンテキストが一覧表示されます。中央の列には、設定されている各コンテキストに対応するコンテキストIDが一覧表示されます。3番目の列には、コンテキストの現在の状態が表示されます。

**ステップ2** 「設定の確認と保存」の章の説明に従って、設定を保存します。

**ステップ3** これで、コンテキストが作成されて、コンテキスト内でインターフェイスと特定の機能を設定できるようになります。特定のサービスとオプションの設定手順については、他のセクションを参照してください。

## イーサネットインターフェイスとポート

アプリケーションインターフェイスのタイプに関係なく、これを作成して設定する手順は次のとおりです。

**ステップ1** [インターフェイスの作成 \(117 ページ\)](#) の設定例を適用して、インターフェイスを作成し、IP アドレスとサブネットマスクを割り当てます。

**ステップ2** [ポートの設定とインターフェイスへのバインド \(117 ページ\)](#) の設定例を適用して、インターフェイスで使用する物理ポートを割り当て、そのポートをインターフェイスにバインドします。

**ステップ3** 必要に応じて、[インターフェイスのスタティックルートの設定 \(118 ページ\)](#) の設定例を適用して、インターフェイスのスタティックルートを設定します。

**ステップ4** 設定するインターフェイスごとに上記の手順を繰り返します。

この項では、システムがネットワーク上で通信できるようにするためのインターフェイスとポートを設定するための最低限の手順について説明します。追加のインターフェイスまたはポートプロパティを設定するコマンドについては、『*Command Line Interface Reference*』の「*Ethernet Port Configuration Mode Commands*」および「*Ethernet Interface Configuration Mode Commands*」の章を参照してください。

システムラインカードとポートレベルの冗長性メカニズムが適切に機能するようにするには、任意のシステムポートに直接接続されているデバイスでスパニングツリープロトコルを無効にする必要があります。

スパンニングツリープロトコルをオフにしないと、冗長性メカニズムまたはサービスの停止に障害が発生する可能性があります。

## インターフェイスの作成

コンテキストで新しいインターフェイスを作成するには、次の例を使用します。

```
configure
  context name
    interface name
      { ip | ipv6 } address address subnetmask [ secondary ]
    end
```

注：

- オプション：インターフェイスのタイプを「loopback」に設定して常にUPにし、物理ポートにバインドしないようにするには、**interface name** コマンドに **loopback** キーワードを追加します。
- オプション：複数のIPアドレスをインターフェイスに割り当てるには、**{ ip | ipv6 } address** コマンドに **secondary** キーワードを追加します。IPアドレスは、IPv4のドット付き10進表記またはIPv6のコロンで区切られた16進表記を使用して入力できます。
- オプション：指定したIPアドレスへの接続が失われた場合に、冗長ラインカード上のポートへのスイッチオーバー用のインターフェイスを設定するには、インターフェイスコンフィギュレーションモードで、**port-switch-on-L3-fail address** コマンドを追加します。このIPアドレスは、IPv4のドット付き10進表記またはIPv6のコロンで区切られた16進表記を使用して入力できます。

## ポートの設定とインターフェイスへのバインド

インターフェイスにポートを設定して割り当てるには、次の設定例を使用します。

```
configure
  port ethernet slot#/port#
    description description
    no shutdown
    bind interface interface_name context_name
  end
```

注：

- オプション：ポート設定を指定する場合は、イーサネットポートのコンフィギュレーションモードで、優先スロット **slot#** コマンドを追加します。
- バインドにより、ポートとそのすべての設定が指名されたインターフェイスに関連付けます。

## インターフェイスのスタティックルートの設定

インターフェイスのスタティックルートを設定するには、次の例を使用します。

```
configure
context name
  { ip | ipv6 } route ip_address netmask next-hop gw_address interface_name
end
```

注：

- *ip\_address* と *netmask* は、ターゲットネットワークの IP アドレスとサブネットマスクです。この IP アドレスは、IPv4 のドット付き 10 進表記または IPv6 のコロンで区切られた 16 進表記を使用して入力できます。
- *gw\_address* は、デフォルトゲートウェイまたはネクストホップルートの IP アドレスです。この IP アドレスは、IPv4 のドット付き 10 進表記または IPv6 のコロンで区切られた 16 進表記を使用して入力できます。
- ゲートウェイルータへのルートを設定するには、ネットワークおよびマスク変数に 209.165.200.225 を使用します。
- 必要に応じて繰り返します。優先ルートに障害が発生した場合に、代替の通信手段を提供するために、複数のスタティックルートを同じ接続先に設定できます。

## ポート設定の表示と確認

**ステップ 1** 次のコマンドを入力して、インターフェイスの設定が正しいことを確認します。

```
[local]host_name# context context_name
[context_name]host_name# show { ip | ipv6 } interface
```

*context\_name* は、インターフェイスが作成されたコンテキストの名前を表します。これらのコマンドの出力は、以下の例のようになります。

この例では、*mgmt1* という名前のインターフェイスがローカルコンテキストで設定されています。

例：

この例では、*mgmt1* という名前のインターフェイスがローカルコンテキストで設定されています。

```
Intf Name:      mgmt1
Intf Type:      Broadcast
IP State:      UP (Bound to 10/11 untagged, ifIndex 285278209)
IP Address:    192.168.100.3      Subnet Mask:    255.255.255.0
Bcast Address: 192.168.100.255    MTU:            1500
Resoln Type:   ARP              ARP timeout:    3600 secs
Number of Secondary Addresses: 0
Total interface count:      1
```

**ステップ 2** 次のコマンドを入力して、ポートの設定が正しいことを確認します。

```
[context_name]host_name# show configuration port slot#/port#
```

例：

このコマンドは、次の例に示すような出力を生成します。これは、シャーシスロット1のポート11の設定を示しています。この例では、ポートは、*source* という名前のコンテキストで設定された *rp1* という名前のインターフェイスにバインドされています。

```
config
  port ethernet 1/11
    description 11_RP1
    no shutdown
    bind interface rp1 source
  end
```

**ステップ3** 次のコマンドを入力して、スタティックルートが正しく設定されていることを確認します。

```
[context_name]host_name# show ip static-route
```

例：

このコマンドでは、次の例に示すような出力が生成されます。これは、209.165.200.226のIPアドレスを持つゲートウェイへのスタティックルートを示しています。

Destination	Nexthop	Protocol	Prec	Cost	Interface
0.0.0.0/0	209.165.200.226	Static	0	0	vNIC1
0.0.0.0/0	209.165.200.226	Static	0	0	rp1 source

**ステップ4** 「設定の確認と保存」の章の説明に従って、設定を保存します。

## VLANs

仮想LAN (VLAN) を使用すると、2つの論理的に分離されたネットワークで同じ物理メディアを使用できます。また、VLAN セグメンテーション (802.1q タグ付けとも呼ばれる) は、VLAN ID を識別するタグを各イーサネットフレームに追加することによって機能します。

特定の packets タイプを処理するための VLAN の作成方法については、「VLAN」の章を参照してください。

## ハイパーバイザ

KVM での VLAN の使用は、ブリッジインターフェイス共有の拡張機能です。違いは、ブリッジセットに参加するインターフェイスです。物理インターフェイス (eth0、eth1 など) は、各ゲストが使用するブリッジにバインドされます。これらのインターフェイスは、VLAN ID タグの有無にかかわらず、外部から着信するか、または内部で生成された未変更の packets を伝送します。

VMware は、1台の vSphere ホスト上の仮想マシンが物理スイッチと同じプロトコルを使用して相互に通信できるようにする仮想スイッチの使用をサポートしています。vSwitch は、データリンク層でフレームを転送することによって、従来の物理イーサネットネットワークスイッチをエミュレートします。vSphere ホストは、仮想マシン用に 1,000 を超える内部仮想ポート

をそれぞれに備えた多数の仮想スイッチを搭載できます。vSphere プラットフォームは、ホストレベルの vSphere 標準スイッチの仮想スイッチ設定と、vSphere 分散型スイッチ（関連付けられた複数のホストを対象とする 1 つの仮想スイッチ）をサポートしています。

## VLAN および管理ポート

管理インターフェイスは、VLAN 設定をサポートしています。このサポートはローカルコンテキストに拡張されます。

バルク統計情報は、通常の管理インターフェイス以外のインターフェイスから送信できます。このインターフェイスは、VLAN もサポートしています。

また、別の VLAN で他の OA&M サービスを設定することもできます。

OA&M サービスに個別の送信元 IP アドレスを割り当てることができます。OA&M サービスは、サービス VLAN と同じ VLAN にバインドすることはできません。サービスには、SGi、Gi、Pi、eGTP またはその他のパケットコア固有のインターフェイスとサービスが含まれます。

この機能は、`vlan` コマンドのサポートをローカルコンテキストの管理ポートに追加することによって実装されます。次のコマンドシーケンスの例を参照してください。

```
configure
  port ethernet 1/1
    vlan 184
    no shutdown
    bind interface 19/3-UHA foo
```





## 第 9 章

# システムセキュリティ

この章では、StarOS のセキュリティ機能について説明します。

この章は、次の内容で構成されています。

- [パスワードの保護 \(121 ページ\)](#)
- [ICSR 設定のサポート \(123 ページ\)](#)
- [暗号化された SNMP コミュニティストリング \(123 ページ\)](#)
- [強化されたパスワードセキュリティ \(123 ページ\)](#)
- [合法的傍受の制約事項 \(123 ページ\)](#)
- [ユーザーの追加、変更、削除 \(124 ページ\)](#)
- [Test-Commands \(125 ページ\)](#)
- [暗号化のための COTS ハードウェアの使用 \(128 ページ\)](#)
- [OS とプラットフォームの乱数生成器のサポート \(129 ページ\)](#)

## パスワードの保護

インスペクタおよびオペレータの権限レベルを持つユーザーは、コマンドラインインターフェイス (CLI) を介して、コンフィギュレーションファイルに復号されたパスワードを表示できません。

## セキュアなパスワードの暗号化

21.0 よりも前の StarOS リリースでは、デフォルトで、システムは MD5 ベースの暗号 (オプション A) を使用してパスワードを暗号化します。これらのパスワードには、パスワードにランダムな 64 ビット (8 バイト) のソルトが追加されています。シャークキーは暗号キーとして使用されます。

シャークキーを設定すると、暗号解読に「共有秘密」の知識が必要な暗号化方式がサポートされます。この共有秘密の知識を持つシャークのみがパスワードにアクセスできます。パスワードを解読するために、シャークキーを知っているハッカーは、暗号化内の 64 ビットのランダムなソルト値の場所を特定する必要があります。

MD-5 で暗号化されたパスワードには、暗号化に使用される方法を識別するために、コンフィギュレーションファイルに「+A」プレフィックスが付いています。



**重要** デフォルトは Algorithm B です。

別のタイプの暗号化アルゴリズムを指定します。グローバル コンフィギュレーション モードの **cli-encrypt-algorithm** コマンドを使用すると、オペレータはパスワード暗号化と秘密暗号化のアルゴリズムを設定できます。21.0 よりも前のリリースでデフォルトの暗号化とパスワードのアルゴリズムは、前述のように MD-5 です (オプション A)。2 番目のパスワード暗号化のアルゴリズム (オプション B) では、暗号化には AES-CTR-128、認証には HMAC-SHA1 が使用されます。暗号キーによってパスワードの機密性が保護され、認証キーによって整合性が保護されます。リリース 21.0 以降では、アルゴリズム B がデフォルトです。このキーで暗号化されたパスワードは、コンフィギュレーションファイルに「+B」プレフィックスが付きます。

3番目のタイプの暗号化アルゴリズムを指定できます (オプション C)。このアルゴリズムは、暗号化と認証に HMAC-SHA512 暗号アルゴリズムの使用を指定します。このキーで暗号化されたパスワードは、コンフィギュレーションファイルに「+C」プレフィックスが付きます。

暗号キーはシャージ ID と内部の乱数生成器から取得した 16 バイトの初期化ベクトル (IV) からハッシュされます。同じ暗号キーと IV のペアを使用して 2 つのパスワードが暗号化されることはありません。シャージ ID と結果の暗号キーを生成するために、セキュリティ管理者はシャージキーを設定する 必要があります。ローカル MAC アドレスに基づくデフォルトのシャージキーはサポートされなくなりました。

**cli-encrypt-algorithm** コマンドのシンタックスは次のとおりです。

```
config
  cli-encrypt-algorithm { A | B | C }
```

## 最新ではない暗号化および復号化のサポート

システムは、以前にフォーマットされた暗号化パスワードをサポートしています。暗号化されたパスワードのシンタックスは、暗号化に使用された方式を示しています。暗号化されたパスワードの前にプレフィックスが表示されない場合は、固定キーを使用した以前の暗号化方式が使用されます。暗号化されたパスワードに「+A」プレフィックスが含まれている場合、復号化方式はシャージキーとランダムなソルトを使用します。

ユーザーが新しい設定を保存すると、生成されたファイルには常に最新の方式で暗号化されたパスワードが含まれます。ユーザーは、以前の DES ベースの暗号化値を生成できません。ただし、将来のすべての StarOS リリースでは、引き続き双方向で暗号化可能なすべてのパスワードのプレーンテキスト パスワード エントリをサポートします。

「ロックアウト」状態を引き起こすことなくシャージキーを変更するための推奨プロセスは、次のとおりです。

- 前のシャージキーを使用して、最後の正常な設定の設定ファイルをロードします。
- 必要な新しい値にシャージキーを変更します。

- この新しいシャードキーマを使用して設定を保存します。

詳細については、「システム設定」の「シャードキーマの設定」を参照してください。

## ICSR 設定のサポート

シャード間セッションリカバリ (ICSR) は、冗長性ペアとして同様に構成された2つの VPC-SI シャードインスタンスを採用する冗長構成です。

ICSR ペアは同じシャードキーマを共有します。2つのシャードインスタンスに互換性のないシャードキーマがあることを ICSR が検出した場合、エラーメッセージがログに記録されますが、ICSR システムは引き続き実行されます。一致するシャードキーマがないと、アクティブペアがアウトオブサービスになった場合に、スタンバイ ICSR ペアはサービスをリカバリできます。スタンバイペアは、暗号化された形式でパスワードに引き続きアクセスできます。

ICSR ペアは、サービス冗長性プロトコル (SRP) を使用して、冗長構成が暗号化されたパスワードまたは DES ベースの双方向暗号化文字列と一致するかどうかを定期的に確認します。設定はソフトウェアに内部的に生成されるため、ユーザーは ICSR の互換性をチェックするために使用される設定にアクセスできません。

## 暗号化された SNMP コミュニティストリング

Simple Network Management Protocol (SNMP) は、ネットワーク要素のパスワードとしてコミュニティストリングを使用します。これらのコミュニティストリングは SNMP PDU のクリアテキストで送信されますが、値はコンフィギュレーションファイルで暗号化できます。

`snmp community encrypted name` コマンドは、SNMP コミュニティストリングの暗号化を有効にします。詳細については、『*Command Line Interface Reference*』の「*Global Configuration Mode Commands*」の章をご覧ください。

## 強化されたパスワードセキュリティ

### 合法的傍受の制約事項

この項では、合法的傍受 (LI) のプロビジョニングに関連付けられている一部のセキュリティ機能について説明します。

### LI サーバーアドレス

外部認証エージェント (RADIUS または Diameter など) は、`access-accept` の一部として LI サーバーアドレスのリストを送信します。そのサブスクリバにすでにインストールされている、

またはこれからインストールされるすべての代行受信については、認証エージェントから受信したすべての LI アドレスと LI サーバーアドレスを照合するためのセキュリティチェックが実行されます。この基準を満たすアドレスのみが、そのサブスクライバの代行受信情報を取得します。

キャンプオントリガーを設定している間、ユーザーは、接続先の LI サーバーアドレスを入力する必要はありません。そのキャンプオントリガーに一致するコールが検出されると、認証エージェントから受信したリストを使用してセキュリティチェックが実行されます。LI 関連の情報は、一致するアドレスが見つかった場合にのみ転送されます。

アクティブのみの代行受信が設定されている場合、一致するコールが検出されると、認証エージェントから受信した LI アドレスに対してセキュリティチェックが行われ、代行受信の設定は拒否されます。

そのサブスクライバの LI サーバーアドレスに関連する情報が受信されない場合、LI サーバーアドレスは制限されません。




---

**重要** 認証エージェントを介して、最大 5 つの LI サーバーアドレスがサポートされます。

---




---

**重要** RADIUS 属性を使用した LI コンテンツおよびイベント配信の接続先アドレスを制限する機能は、PDSN および HA ゲートウェイでのみサポートされています。

---

## 代行受信の変更

1 人の LI 管理者は、別の LI 管理者が作成した代行受信にアクセスしたり、変更したりできます。代行受信が追加、削除、または変更されるたびに、変更に関するイベントログがすべての LI 管理者に表示されます。また、SNMP トラップも生成されます。

## ユーザーの追加、変更、削除

システムに対するユーザーの追加または削除は一般には行われません。同じように、ユーザーの権限を変更することもまれです。ただし、システムが侵害された場合は、攻撃者が特権ユーザーを追加または削除したり、権限を上げたり、他のユーザーの権限を低くしたりすることがよくあります。

一般的なルールとして、権限の低いユーザーが権限を上げたり、権限の高いユーザーが入力したパスワードなどのセンシティブデータにアクセスしたりできないようにする必要があります。




---

**重要** システムは、これらのユーザーがシステムを介して設定されている場合にのみ、ユーザーおよびユーザー属性（権限レベルなど）の変更を検出できます。

---

## ユーザーの追加または削除の通知

承認レベルの低いユーザーは、承認レベルの高いユーザーを作成できません。ただし、悪意のある攻撃者が高レベルの権限を持つユーザーを作成できる場合は、他の高レベルの認可されたユーザーを削除して、システムからロックアウトすることができます。

次の SNMP トラップは、ユーザーが追加または削除されたときに管理者に通知します。

- **starLocalUserAdded** : 新しいローカルユーザーアカウントがシステムに追加されたことを示します。
- **starLocalUserRemoved** : ローカルユーザーアカウントがシステムから削除されたことを示します。

## 権限レベル変更の通知

ユーザーの権限レベルが高くなるか低くなるたびに、SNMP 通知が送信されます。悪意のある攻撃者は、権限を何らかの方法で「昇格」させることで、より多くの権限を持つコマンドにアクセスする可能性があります。これが実行されると、他のすべてのユーザーの権限を「降格」させることができ、その結果、適切な管理者がシステムからロックアウトされます。

**starLocalUserPrivilegeChanged** トラップは、ローカルユーザーの権限レベルが変更されたことを示します。

## オペレーティング システム シェルへのユーザーアクセス

**starOsShellAccessed** トラップは、ユーザーがオペレーティングシステムのシェルにアクセスしたことを示します。

## Test-Commands

セキュリティ管理者または管理者権限を持つユーザーは、以前に非表示になっていた `test-commands` の表示を有効にすることができます。CLI `test-commands` では、既存のコマンドの新しいコマンドキーワード、および新しいコマンドが表示されます。



**注意** CLI `test-commands` は、診断の使用のみを目的としています。通常のシステム操作では、これらのコマンドへのアクセスは必要ありません。これらのコマンドは、Cisco TAC の担当者のみが使用することを目的としています。これらのコマンドの一部は、システムパフォーマンスの低下、サブスクライバの削除、および/またはシステムの動作不能の原因となることがあります。

## cli test-commands モードの有効化

test-commands へのアクセスを有効にするには、セキュリティ管理者がグローバルコンフィギュレーションモードにログインし、**cli hidden** を入力する必要があります。

次に、このコマンドシーケンスを示します。

```
[local]host_name# config
[local]host_name(config)# cli hidden
[local]host_name(config)#
```

デフォルトでは、**cli hidden** は無効になっています。



**重要** 低レベルの診断およびテストコマンドやキーワードは、管理者または上位の権限を持つユーザーに表示されるようになりました。CLIには、test-commands モードが有効になっていることを示すビジュアルインジケータはありません。

## CLI テストコマンドにアクセスするためのパスワードの有効化

セキュリティ管理者は、CLI テストコマンドにアクセスするためのプレーンテキストのパスワードまたは暗号化されたパスワードを設定できます。*password* の値は、起動設定情報とともに **/flash** に保存されます。**show configuration** コマンドと **save configuration** コマンドがこの値をプレーンテキストで出力することはありません。

グローバルコンフィギュレーションモードコマンドの **tech-support test-commands [encrypted] password new\_password [ old-password old\_password ]** は、CLI テストコマンドにアクセスするための暗号化されたパスワードまたはプレーンテキストのパスワードを設定します。

次に、このコマンドシーケンスを示します。

```
[local]host_name# config
[local]host_name(config)# tech-support test-commands password new_password [
old-password old_password ]
[local]host_name(config)#
```

新しいパスワードで既存のパスワードを置き換える場合は、変更を受け入れるために古いパスワードを入力する必要があります。

古いパスワードが入力されていない場合、または既存の設定値と一致しない場合は、「tech-support password is already configured」というエラーメッセージが表示されます。次に、古いパスワードのエントリを受け入れるように求めるプロンプト（「Enter old tech-support password:」）が表示されます。

**old-password old\_password** を入力することで、古いパスワードの入力を求められることなく、既存のパスワードを置き換えることができます。古いパスワードを誤って入力した場合、または古いパスワードを入力しなかった場合は、「Failure: Must enter matching old tech-support password to replace existing password」というエラーメッセージが表示されます。

クイックセットアップウィザード（Exec モードの **setup** コマンド）でも、テクニカルサポートのテストコマンドパスワードの入力が求められます。古いテクニカルサポートのパスワード

を忘れた場合は、コンソールポートから直接 `setup` を実行して、新しいテクニカルサポートパスワードを入力できます。

テストコマンドのパスワードが設定されている場合、グローバルコンフィギュレーションモードコマンドの `cli test-commands [ encrypted ] password password` にはパスワードのキーワードを入力する必要があります。 `encrypted` キーワードを指定した場合、 `password` 引数は、パスワード値を含む暗号化された文字列として解釈されます。 `encrypted` キーワードを指定しなかった場合、 `password` 引数は実際のプレーンテキスト値として解釈されます。



---

**重要** `tech-support test-commands password` が設定されていない場合は、StarOS が新しいパスワードを作成します。 `cli test-commands` に `password` キーワードを入力しなかった場合は、パスワードの入力が求められます（エコーなし）。また、CLI テストコマンドにアクセスするには、管理者が `cli hidden` を有効にする必要があります。

---

## Exec モードの `cli test-commands`

Exec モードコマンドは、Exec モードから `cli` テストコマンドを入力する特権ユーザーが使用できます。

```
[local]host_name# cli test-commands [encrypted] password password
Warning: Test commands enables internal testing and debugging commands
USE OF THIS MODE MAY CAUSE SIGNIFICANT SERVICE INTERRUPTION
```



---

**重要** ユーザーがCLIテストコマンドモードを開始するたびに、SNMPトラップ（`starTestModeEntered`）が生成されます。

---

## コンフィギュレーションモードの `cli test-commands`

低レベルのソフトウェアパラメータにアクセスできるようにするためのコンフィギュレーションコマンドには、権限を持つユーザーがグローバルコンフィギュレーションモードでコマンド `cli test-commands` を入力した後にのみアクセスできます。

```
[local]host_name# config
[local]host_name(config)# cli test-commands [encrypted] password password
Warning: Test commands enables internal testing and debugging commands
USE OF THIS MODE MAY CAUSE SIGNIFICANT SERVICE INTERRUPTION
```



---

**重要** ユーザーがCLIテストコマンドモードを開始するたびに、SNMPトラップ（`starTestModeEntered`）が生成されます。

---

## 暗号化のための COTS ハードウェアの使用

StarOS VPC インスタンスは、ソフトウェア内でパケットの暗号化とトンネリングを実行します。ただし、市販（COTS）のサーバーが Intel Communications のチップセット 89xx を使用し、このチップセットにパススルーするように VPC 仮想マシンを設定した場合、VPC インスタンスは自動的にこのハードウェアチップを使用してパケットの暗号化と復号を実行します。Intel Communications のチップセット 89xx は、Coletto Creek とも呼ばれています。



- (注) VPC が暗号化と復号にハードウェアチップセットを使用するためには、サービス機能 (SF) VM のすべてで Intel Communications のチップセットを使用する必要があります。

COTS サーバーがこのチップセットを使用しているかどうかを確認するには、**show hardware** コマンドを使用してすべてのスロットの情報を表示します。次に、Coletto Creek 暗号化アクセラレータを使用するハードウェア上の VPC SI インスタンスに対する **show hardware** コマンドの出力例を示します。

```
[local]swch32# show hardware
System Information:
  Platform           : KVM Guest
  UUID/Serial Number : 014A4D4F-7644-4CF1-C408-8ABB631B3E34
  CPU Packages       : 1 [#0]
  CPU Nodes          : 1
  CPU Cores/Threads  : 16
  Memory             : 16384M (qvmc-si-medium)
  Crypto Accelerator : Coletto Creek A0
Storage Devices:
  Virtual Flash      : Present
  Type               : 4096M disk
  Model              : ATA-QEMUHARDDISK
  Serial Number      : QM00001
  Hard Drive 1       : Present
  Type               : 16384M disk
  Model              : ATA-QEMUHARDDISK
  Serial Number      : QM00002
  Hard Drive 2       : Not Present
  USB 1              : Not Present
  USB 2              : Not Present
  CDROM 1           : Present
  Type               : cdrom
  Model              : QEMU-QEMUDVD-ROM
Network Interfaces:
  loeth0  addr 52:54:00:ae:b7:72 at virtio1, laf4:0001 (virtio_net)
  RxQ(s)/RINGSZ/COALESCE: 1/256/-NA-
  TxQ(s)/RINGSZ/COALESCE: 1/256/-NA-
  NODE-ID           : -NA-
  port1_10 addr 00:1b:21:87:14:ac at 0000:00:06.0, 8086:10fb (ixgbe)
  RxQ(s)/RINGSZ/COALESCE: 16/4096/500
  TxQ(s)/RINGSZ/COALESCE: 16/4096/0
  NODE-ID           : -NA-
  port1_11 addr 00:1b:21:87:14:ad at 0000:00:07.0, 8086:10fb (ixgbe)
  RxQ(s)/RINGSZ/COALESCE: 16/4096/500
  TxQ(s)/RINGSZ/COALESCE: 16/4096/0
  NODE-ID           : -NA-
```



次に、暗号アクセラレータがインストールされていないハードウェア上の VPC SI インスタンスに対する **show hardware** コマンドの出力例を示します。

```
[local]swch81# show hardware
System Information:
Platform           : KVM Guest
UUID/Serial Number : E0A26495-F822-4AC0-914D-B51332177C4D
CPU Packages       : 1 [#0]
CPU Nodes          : 1
CPU Cores/Threads  : 16
Memory             : 32768M (qvpc-si-medium)
Crypto Accelerator : None
Storage Devices:
Virtual Flash      : Present
  Type             : 4096M disk
  Model            : ATA-QEMUHARDDISK
  Serial Number    : QM00001
Hard Drive 1       : Present
  Type             : 16384M disk
  Model            : ATA-QEMUHARDDISK
  Serial Number    : QM00002
Hard Drive 2       : Not Present
USB 1              : Not Present
USB 2              : Not Present
CDROM 1            : Present
  Type             : cdrom
  Model            : QEMU-QEMUDVD-ROM
Network Interfaces:
loeth0  addr 52:54:00:e9:70:05 at virtio1, 1af4:0001 (virtio_net)
  RxQ(s)/RINGSZ/COALESCE: 1/256/-NA-
  TxQ(s)/RINGSZ/COALESCE: 1/256/-NA-
  NODE-ID                : -NA-
port1_10 addr 52:54:00:22:f7:85 at virtio2, 1af4:0001 (virtio_net)
  RxQ(s)/RINGSZ/COALESCE: 8/256/-NA-
  TxQ(s)/RINGSZ/COALESCE: 8/256/-NA-
  NODE-ID                : -NA-
port1_11 addr 52:54:00:3e:67:f9 at virtio3, 1af4:0001 (virtio_net)
  RxQ(s)/RINGSZ/COALESCE: 8/256/-NA-
  TxQ(s)/RINGSZ/COALESCE: 8/256/-NA-
  NODE-ID                : -NA-
```

## OS とプラットフォームの乱数生成器のサポート

### 機能の概要と変更履歴

#### 要約データ

該当製品または機能エリア	すべて
該当プラットフォーム	<ul style="list-style-type: none"> <li>• VPC-DI</li> <li>• VPC-SI</li> </ul>
機能のデフォルト	無効：設定が必要

このリリースでの関連する変更点	N/A
関連資料	<ul style="list-style-type: none"> <li>• VPC-DI システム管理ガイド</li> <li>• VPC-SI System アドミニストレーション ガイド</li> </ul>

### マニュアルの変更履歴



**重要** リリース 21.2 および N5.1 よりも前に導入された機能の改訂履歴の詳細は示していません。

改訂の詳細	リリース
最初の導入。	21.13

## 機能説明

ASR 5500 および VPC プラットフォームに展開されているいくつかの機能には、特定のタスクを実行するための乱数が必要です。これらのタスクにはカーネル乱数生成器が使用されていますが、生成される数値は、セキュリティ標準規格に従って十分にランダムである場合とそうでない場合があります。ただし、ハードウェアまたはホストが提供する乱数は、信頼性が高く、セキュリティ標準規格に準拠していると見なされます。

OS およびプラットフォームの乱数生成器のサポート機能は、このセキュリティコンプライアンス要件に対応しています。これにより、システム管理者は、ホストマシン上でハードウェア乱数生成器 (HWRNG) を設定できます。

設定されている場合、システムはハードウェアの乱数生成器を使用します。



(注) この機能は、HWRNG サポートがホストで使用可能な場合にのみ機能します。

HWRNG サポートが使用可能な場合は、ホスト上の libvirt xml ファイルに次の設定を追加します。これにより、クライアント (StarOS) に virtio\_rng サポートが追加されます。

```
<rng model='virtio'>
  <backend model='random'/>/dev/random</backend>
  <address type='pci' domain='0x0000' bus='0x00' slot='0x07' function='0x0' />
</rng>
```



(注) (前述の設定に示すように) 設定でスロット番号7を使用して競合がある場合、次に使用可能なスロットを使用します。

この設定は、それぞれの展開設定に基づいてサポートされているプラットフォームに適用する必要があります。

クライアントでの設定変更は必要ありません。クライアント (StarOS) は、ホストでサポートが有効になっている場合、自動的に `virtio_rng` を選択します。





## 第 10 章

# セキュアなシステムコンフィギュレーションファイル

- [機能の概要と変更履歴 \(133 ページ\)](#)
- [機能説明 \(134 ページ\)](#)
- [システム コンフィギュレーション ファイルの保護方法 \(134 ページ\)](#)
- [署名検証の設定 \(135 ページ\)](#)

## 機能の概要と変更履歴

### 要約データ

該当製品または機能エリア	すべて
該当プラットフォーム	ASR 5500 VPC-DI VPC-SI
機能のデフォルト	無効
このリリースでの関連する変更点	N/A
関連資料	<ul style="list-style-type: none"><li>• <i>ASR 5500 System Administration Guide</i></li><li>• <i>VPC-DI システム管理ガイド</i></li><li>• <i>VPC-SI System アドミニストレーションガイド</i></li></ul>

### マニュアルの変更履歴

改訂の詳細	リリース
最初の導入。	21.3

## 機能説明

システム設定ファイルには、オペレータのネットワークのセットアップと運用に使用される重要な設定情報が含まれています。設定ファイルは、ネットワークに悪影響を及ぼす可能性のあるファイルへの不正な変更を避けるために、ロード前に適切に承認されている必要があります。

この機能により、ロードされる前にコンフィギュレーションファイルの整合性と信頼性を確保するために、システム設定ファイルを RSA キーで署名できるようになります。オペレータは秘密キーを使用して、コンフィギュレーションファイルに署名できます。システムは公開キーを使用して、署名されたコンフィギュレーションファイルをロードする前に検証します。

## システム コンフィギュレーション ファイルの保護方法

### デジタル署名の作成

オペレータは、次の手順を使用して設定ファイルに署名できます。

1. 設定ファイル上で SHA512 ハッシュを実行して、メッセージダイジェストを作成します。

例 (Linux/OpenSSL) :

```
openssl dgst -sha512 -binary -out digest cfg_file
```

2. RSA 秘密キーを使用してメッセージダイジェスト値を暗号化することによって、デジタル署名を作成します。

例 (Linux/OpenSSL) :

```
openssl pkeyutl -sign -in digest -inkey pri_key.pem -out sig \  
-pkeyopt digest:sha512 -pkeyopt rsa_padding_mode:pss \  
-pkeyopt rsa_pss_saltlen:-2
```

3. デジタル署名を base64 形式に変換します (先頭に「#」が追加され、最後に新しい行が追加されます)。

例 (Linux/OpenSSL) :

```
echo -n "#" > sig_base64  
base64 sig -w 0 >> sig_base64  
echo "" >> sig_base64
```

4. デジタル署名を使用して元の設定ファイルを追加します。

例 (Linux/OpenSSL) :

```
cat sig_base64 cfg_file > signed_cfg_file
```

### 公開キーと秘密キーの生成

RSA 公開キーは PEM 形式 (.pem ファイル) に保存され、次の例の OpenSSL コマンドのいずれかを使用して生成できます。

```
openssl rsa -in pri_key.pem -pubout -out pub_key.pem  
--or--  
openssl rsa -in pri_key.pem -RSAPublicKey_out -out pub_key.pem
```

PEM 形式の RSA 秘密キーは、次の例の OpenSSL コマンドを使用して生成できます。

```
openssl genrsa -out pri_key.pem 2048
```

**openssl rsa** コマンドと **openssl genrsa** コマンドの詳細については、それぞれの OpenSSL のマニュアルページを参照してください。

## デジタル署名の検証

署名の検証が有効になっている場合、システムが起動し、コンフィギュレーションファイル（または設定ファイルがロードされた時点）をロードするときに、デジタル署名の検証が行われます。システムは、セキュアディレクトリで `enable_cfg_pubkey` ファイルを検索して、署名検証が有効（または無効）になっているかどうかを判断します。詳細については、[署名検証の有効化または無効化（136 ページ）](#) を参照してください。

システムは、次の手順を使用して、署名されたコンフィギュレーションファイルを検証します。

1. フラッシュから RSA 公開署名キーを抽出します。
2. コンフィギュレーションファイルのデジタル署名（最初の行）を抽出します。
3. base64 形式からバイナリ形式に署名を変換します。
4. RSA 公開キーを使用して署名を復号します。
5. プレーンコンフィギュレーションファイルの SHA512 ハッシュを計算して、メッセージダイジェストが生成されるようにします。
6. 復号された署名の値と新しく計算されたメッセージダイジェストを比較します。一致する場合、コンフィギュレーションファイルは正常に検証されます。

## 署名検証の設定

### 検証用の RSA 公開キーのインポート

署名済みの設定ファイルを確認するには、RSA 公開キー（PEM 形式）をインポートする必要があります。RSA 公開キーをインポートするには、次のコマンドを使用します。



---

**重要** このコマンドは、コンソールからのみ実行できます。

---

```
cfg-security import public-key url url_address
```

注：

- コマンドが実行されると、既存の .pem ファイルが新しい .pem ファイルに置き換えられます。
- `url_address` はローカルファイルまたはリモートファイルを参照します。また、次の形式を使用して入力する必要があります。

```
[file:]{/flash | /usb1 | /hd-raid | /sftp}{/directory}/filename
```

```
tftp://host[:port][/<directory>]/filename
```

```
ftp://[username[:password]@]host[:port][/directory]/filename
```

```
sftp://[username[:password]@]host[:port][/directory]/filename
```

```
http://[username[:password]@]host[:port][/directory]/filename
```

```
https://[username[:password]@]host[:port][/directory]/filename
```

## 署名検証の有効化または無効化

コンフィギュレーションファイルで署名の検証を有効化（または無効化）するには、次のコマンドを使用します。



---

**重要** このコマンドは、コンソールからのみ実行できます。

---

```
[ no ] cfg-security sign
```

注：

- 署名の検証 (**cfg-security sign** コマンド) を有効にすると、PEM ファイルが存在するディレクトリ内に `enable_cfg_pubkey` という名前の空の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の署名の検証を無効にするには、**no cfg-security sign** コマンドを使用します。署名の検証 (**no cfg-security sign** コマンド) を無効にすると、`enable_cfg_pubkey` ファイルが削除されます。
- システムは、署名の検証が有効か無効かを判断するために、`enable_cfg_pubkey` ファイルを検索します。





## 第 11 章

# ソフトウェア管理の運用

この章では、システム上のソフトウェア管理の運用について説明します。

- ローカルファイルシステムの概要 (137 ページ)
- ローカルファイルシステムの保守 (138 ページ)
- Elastic Services Controller のクラウド初期化サポート (143 ページ)
- 起動スタックの設定 (143 ページ)
- オペレーティングシステムソフトウェアのアップグレード (146 ページ)
- ライセンスキーの管理 (172 ページ)
- ローカルユーザー管理アカウントの管理 (175 ページ)

## ローカルファイルシステムの概要

VPCVM上のローカルファイルシステムは、次のものに保存されているファイルで構成されません。

- **/flash** ハイパーバイザを介して M 上の vHDD-1 として割り当てられたフラッシュメモリは、システムで使用される StarOS イメージ、CLI 設定、およびクラッシュログファイルのデフォルトのストレージメディアです。
- **/hd-raid** これは、ハイパーバイザによって CF VM 上で vHDD-2 として割り当てられたストレージ領域です。CDR (課金データレコード) と UDR (使用データレコード) を保存するために使用されます。

## ローカルファイルシステムで使用されるファイルタイプ

ローカルファイルシステムには、次のファイルタイプを配置できます。

- **オペレーティングシステムソフトウェアのイメージファイル** : このバイナリファイルタイプは **.bin** 拡張子で識別されます。ファイルは、起動時またはリロード時にシステムによってロードされるオペレーティングシステムです。これは、エンドユーザーが変更できない実行可能な読み取り専用ファイルです。

- **CLI 設定ファイル**：このファイルタイプは **.cfg** 拡張子によって識別されます。これらは、オペレーティングシステムのソフトウェアイメージと連携して動作する CLI コマンドを含むテキストファイルです。これらのファイルによって、提供されるサービス、ハードウェアとソフトウェアの設定、システムによって実行されるその他の機能が決まります。通常、ファイルはエンドユーザーが作成します。ファイルはオンラインでも、オフラインでも変更でき、そのファイルを説明する長いファイル名を使用します。
- **システムファイル**：システムでは、**.sys** 拡張子によって識別されるファイル 1 つのみが使用されます。boot.sys ファイルには、システムの検出方法や、起動スタックからファイルグループ（.bin ファイルと .cfg ファイルのペア）をロードする優先順位を説明するシステム固有の情報が含まれています。
- **クラッシュログの要約**：ファイル名の **crashlog** で識別されるクラッシュログの要約には、システムで発生したソフトウェアまたはハードウェアの障害に関するサマリー情報が含まれています。このファイルは、デバイスの **/flash/crsh2/** ディレクトリにあります。CLI を使用してこのファイルの内容を表示することはできますが、ファイルを変更することはできません。

## boot.sys ファイルの概要

システムは boot.sys ファイルを使用して、システムが起動時に使用する優先順位付けされた起動スタックパラメータとファイルグループを保存します。このファイルは、外部手段ではなく、システム CLI コマンドを使用してのみ変更できます。起動パラメータには、オペレーティングシステムのイメージファイルを見つけるために必要な次のような情報が含まれています。

- **bootmode**：この設定は通常は **normal** に設定され、システムの起動方法を識別します。
- **起動スタック情報**：起動スタックは、オペレーティングシステムのイメージファイルと、ロードする CLI 設定ファイルを指定する、優先順位付けられたファイルグループエントリから構成されます。

システムを初めて起動したときに、通常の起動モードを使用して、**/flash** ディレクトリからオペレーティングシステムのソフトウェアイメージをロードするように、boot.sys ファイルが設定されます。

ローカルファイルシステムには CLI 設定ファイルが含まれていません。これにより、システムは最初に正常に起動したときに自動的に CLI ベースのクイック セットアップ ウィザードを開始します。クイックセットアップウィザードの使用法の詳細については、「使用する前に」を参照してください。

## ローカルファイルシステムの保守

ローカルファイルシステムを構成するデバイスを管理および保守するには、CLI コマンドを使用します。この項で説明されているすべてのコマンドは Exec モードで実行します。特に指定がない限り、これらのコマンドを実行するには、セキュリティ管理者または管理者の権限が必要です。

## ファイルシステム管理コマンド

ローカルファイルシステムを管理および構成するには、この項のコマンドを使用します。



---

**重要** 次に示すコマンドの詳細については、『*Command Line Interface Reference*』の「*Exec Mode Commands*」の章を参照してください。

---

### ディレクトリの作成

特定のローカルデバイスに新しいディレクトリを作成するには、**mkdir** コマンドを使用します。このディレクトリは、ローカルファイルシステムにある任意のファイルのパス名の一部として組み込むことができます。

```
[local]host_name# mkdir { /flash | /usb1 | /hd-raid } /dir_name
```

次のコマンドを使用して、*configs* という名前のディレクトリを作成します。

```
[local]host_name# mkdir /flash/configs
```

### ファイルとディレクトリの名前の変更

ファイルの名前を元の名前から別の名前に変更するには、**rename** コマンドを使用します。必要に応じて同じファイル拡張子を使用し、ファイルタイプが変更されないようにします。

*/flash* ローカルデバイス上でファイル名 *iot\_test.cfg* を *iot\_accept.cfg* に変更するには、次のコマンドを使用します。

```
[local]host_name# rename /flash/iot_test.cfg /flash/iot_accept.cfg  
-noconfirm
```



---

**重要** **rename** コマンドは、同じローカルデバイス内でのみ使用します。ファイル名を変更して、そのファイルを別のローカルデバイスに同時に配置することはできません。名前を変更したファイルを移動するには、**copy** コマンドを使用する必要があります。

---

### ファイルのコピー

これらの手順は、Execモードのルートプロンプトを使用していることが前提になります。現在の設定を保存するには、次のコマンドを入力します。

```
[local]host_name# copy from_url to_url [-noconfirm]
```

*system.cfg* という設定ファイルを *cfgfiles* と呼ばれていたディレクトリから *configs\_old* というディレクトリにコピーするには、次のコマンドを入力します。

```
[local]host_name# copy /flash/cfgfiles/system.cfg  
/flash/configs_old/system_2011.cfg
```

`init_config.cfg` という設定ファイルをホスト名が `config_server` の TFTP サーバーのルートディレクトリにコピーするには、次のコマンドを入力します。

```
[local]host_name# copy /flash/cfgfiles/init_config.cfg
tftp://config_server/init_config.cfg
```

## ファイルの削除

**delete** コマンドは、指定されたファイルをローカルファイルシステム上の指定された場所から削除します。



**重要** このコマンドは、ワイルドカードエントリをサポートしていません。各ファイル名は、全体で指定する必要があります。



**注意** `boot.sys` ファイルは削除しないでください。削除すると、システムはコマンドを再起動せず、動作不能になります。

```
[local]host_name# delete { /flash | /usb1 | /hd-raid }/filename [ -noconfirm
]
```

次のコマンドは、`/flash` ディレクトリから `est.cfg` という名前のファイルを削除します。

```
[local]host_name# delete /flash/test.cfg
```

## ディレクトリの削除

**rmdir** コマンドは、特定のローカルデバイス上の現在のディレクトリを削除します。このディレクトリは、ローカルファイルシステムにある任意のファイルのパス名の一部として組み込むことができます。



**重要** 削除するディレクトリは、**rmdir** コマンドを実行する前に空にしておく必要があります。ディレクトリが空でない場合、CLI には「Directory not empty」というメッセージが表示され、実行されません。

```
[local]host_name# rmdir url /dir_name
```

次に、`/flash` ディレクトリ内の `config` という名前の空のディレクトリを削除するコマンドを示します。

```
[local]host_name# rmdir /flash/configs
```

## ローカルデバイスのフォーマット

**format** コマンドは、ローカルデバイスの低レベルフォーマットを実行します。この操作は、FAT16 フォーマット方式を使用するようにデバイスをフォーマットします。これは、オペレーティングシステムの適切な読み取り/書き込み機能に必要です。



**重要** NTFS や FAT32 などの他の方式を使用してフォーマットされたローカルデバイスは、さまざまなオペレーティングシステム、CLI 設定、およびクラッシュログファイルを保存するために使用される場合があります。ただし、通常使用のために新しいローカルデバイスを MIO/UMIO/MIO2 に配置する場合は、使用する前にシステムを介してデバイスをフォーマットする必要があります。これにより、適切なファイルアロケーションテーブルのフォーマットが使用され、他のオペレーティングシステムで使用されている他のフォーマットとの不一致を防止することができます。



**注意** `filesystem format` コマンドは、デバイスに保存されているすべてのファイルと情報を削除します。

ローカルファイルシステムで使用するローカルデバイスをフォーマットするには、次のコマンドを入力します。

```
[local]host_name# filesystem format { /flash | /usb1 | /hd-raid }
```

## 既存の CLI コンフィギュレーション ファイルの適用

既存の CLI 設定ファイルは、ユーティリティ機能（テスト時のすべての統計情報のクリアなど）を提供するために作成されたか、またはテキストエディアを使用してオフラインで作成された `.conf` ファイルです。既存の設定ファイルはローカルファイルシステムに保存されていることがあり、そのファイルはいつでも実行中のシステムに適用できます。



**注意** 現在別の CLI 設定を実行しているシステムに設定ファイルを適用したすると、同じコマンドが適用しようとしている設定ファイルに存在する場合は、類似するコンテキスト、論理インターフェイス、物理ポート、IP アドレス、またはその他の設定済みの項目はすべて上書きされます。適用しようとしているファイルの内容を十分に理解し、現在実行中のコマンドが上書きされた場合はサービスにどのような影響があるかを把握しておく必要があります。また、変更は自動的に保存されないことにも注意してください。

CLI 設定ファイル、または CLI コマンドを含むスクリプトは、Exec モードプロンプトで次のコマンドを入力することによって実行中のシステムに適用できます。

```
[local]host_name# configure url [ verbose ]
```

`url` は、適用する CLI 設定ファイルの場所を指定します。ローカルファイルまたはリモートファイルを参照する場合があります。

次のコマンドは、`/flash` ディレクトリ内の `clearcmds` という名前の既存の CLI 設定ファイルを適用します。

```
[local]host_name# configure /flash/clearcmds.cfg
```

## ローカルファイルシステム上のファイルの表示

この項では、さまざまなファイルを表示する方法について説明します。

### ローカルデバイスの内容の表示

任意のローカルデバイスのコンテンツ、使用状況情報、およびファイルシステムディレクトリ構造を表示するには、Exec モードプロンプトで次のコマンドを入力します。

```
directory { /flash | /usb1 | /hd-raid }
```

### CLI 設定と boot.sys ファイルの表示

ローカルファイルシステムに格納されている CLI 設定ファイルと boot.sys ファイルの内容は、Exec モードのプロンプトで次のコマンドを入力することによって、オフラインで表示できます (OS にロードする必要はありません)。

```
[local]host_name# show file url { /flash | /usb1 | /hd-raid } filename
```

ここで、*url* はファイルの場所のパス名、*filename* は拡張子を含むファイルの名前を指します。




---

**重要** オペレータレベルのユーザーとインスペクタレベルのユーザーは **show file** コマンドを実行できますが、**directory** コマンドを実行することはできません。

---

### オペレーティングシステムのファイルの検証

.bin 拡張子で識別されるオペレーティングシステムのソフトウェアイメージファイルは、システム上で実行され、ランタイムオペレーティングシステム (OS) を作成する、読み取り不可能かつ編集不可能なファイルです。

新しいオペレーティングシステムイメージファイルをロードする前に確認することが重要です。これを実現するために、独自のチェックサムアルゴリズムを使用して、プログラムのコンパイル中に .bin ファイル内に保存されているアプリケーションの各部分のチェックサム値を作成します。

この情報は、コンパイル中にファイル内に保存されているチェックサム値に対して実際のファイルを検証するために使用できます。イメージファイルのいずれかの部分が破損した場合 (たとえば、ファイルが切り捨てられた場合や、バイナリモードではなく ASCII モードを使用して転送された場合など)、この情報が報告され、ファイルは使用できないと見なされます。

オペレーティングシステムのソフトウェアイメージファイルを検証するには、Exec モードプロンプトで次のコマンドを入力します。

```
[local]host_name# show version { /flash | /usb1 | /hd-raid }  
/[directory]/filename [all]
```

このコマンドの出力には、次の情報の列が表示されます。

- バージョン番号

- 説明
- 日付 (Date)
- 起動イメージ
- サイズ
- フラグ
- プラットフォーム

無効なファイルが見つかった場合、システムは次のようなエラーメッセージを表示します。

```
Failure: Image /flash/image_version.bin CRC check failed!  
Failure: /flash/image_version.bin, has a bad magic number
```

## Elastic Services Controller のクラウド初期化サポート

Elastic Services Controller (ESC) が QvPC-DI の制御機能 (CF) にアクティブおよびスタンバイで Cinder マルチアタッチボリュームを使用する場合、仮想マシン (VM) オーケストレーション用に呼び出される Openstack API バージョンは 2.60 以上です。この API バージョンでは、ESC はセキュリティ上の理由から、構成ファイルをエンコードして VM に挿入します。VM はエンコードされた構成ファイルを読み取ることができないため、ESC は **user\_data** 圧縮ファイルを使用します。この **user\_data** ファイルには、VM の起動に必要な構成ファイルが含まれています。

## 起動スタックの設定

起動スタックは、オペレーティングシステムのソフトウェアイメージと CLI 設定ファイルとの関連付けに優先順位を付けたリストで構成されます。これらの関連付けによって、システムの起動時またはリロード/再起動時にロードされるソフトウェアイメージと設定ファイルが決まります。関連付けは複数設定できますが、システムが使用するのは最も高い優先順位を持つ関連付けです。この関連付けの処理中にエラーが発生した場合 (いずれかのファイルが見つからないなど)、システムは次に高い優先順位の関連付けを使用しようとします。

VPC-SI プラットフォームと VPC-DI プラットフォームでは、設定されたブート優先順位が最も高い構成ファイルが利用できない場合 (ただし、イメージファイルは利用可能)、システムは次に利用可能なブートシステムの優先順位を使用する代わりに、リロード後に設定ウィザードを使用して起動します。優先順位は 1 ~ 100 で、1 が最も高い優先順位です。boot.sys ファイル内に設定できる起動スタックエントリの最大数は 10 です。

起動スタック情報は、[boot.sys ファイルの概要 \(138 ページ\)](#) で説明されているように、boot.sys ファイルに含まれています。次の項で説明するように、boot.sys ファイルには、起動スタックエントリの他に、システムの起動方法を定義するために必要なすべての設定コマンドが含まれています。

## システムの起動方式

ローカル起動方式では、システムにローカルに保存されているソフトウェアイメージと設定ファイルを使用します。システムの起動時または再起動時に、ローカルデバイスのいずれか、または **/hd-raid** で特定のソフトウェアイメージとそれに付随する設定テキストファイルを調べます。ローカル起動方式を使用している場合は、起動スタックパラメータの設定のみが必要です。

システムは、特定の外部ネットワークサーバーからシステムに存在する設定テキストファイルとペアになっているソフトウェアイメージを取得します。ネットワーク起動を使用する場合は、次を設定する必要があります。

- 起動スタックパラメータ。これらは使用するファイルとそれらに使用する優先順位を定義します。
- リモート管理 LAN インターフェイスを定義する起動インターフェイスおよびネットワークパラメータと、外部ネットワークサーバーに到達するために使用する方式
- 確立するネットワーク通信を可能にするための遅延期間（秒単位）を定義するネットワーク起動遅延時間およびオプションのネームサーバーパラメータと、使用される可能性があるドメインネームサービス（DNS）ネームサーバーの IP アドレス

## 現在の起動スタックの表示

boot.sys ファイルに含まれている起動スタックのエントリを表示するには、Exec モードの **show boot** コマンドを実行します。



**重要** **show boot** コマンドはオペレータレベルのユーザーとインスペクタレベルのユーザーが実行できます。

次に、ローカル起動設定のコマンド出力の例を示します。これらの例では、イメージファイル（オペレーティングシステムソフトウェア）と設定ファイル（CLI コマンド）の両方が、**/flash** デバイ스에配置されていることに注意してください。



**重要** StarOS イメージファイル名の形式は「**asr5500-image\_number.bin**」です。

例：

```
boot system priority 18 \
  image /flash/16-1-builds/asr5500-16.1.3.bin \
  config /flash/general_config.cfg

boot system priority 19 \
  image /flash/16-1-builds/asr5500-16.1.1.bin \
  config /flash/general_config_3819.cfg

boot system priority 20 \
```



```
image /flash/16-1-builds/asr5500-16.1.0.bin \  
config /flash/general_config_3665.cfg
```

次に、ネットワーク起動とローカル起動を組み合わせた設定の出力の例を示します。この例では、イメージファイル（オペレーティングシステムソフトウェア）は最初の2つの起動スタックエントリ（優先順位 18 と 19）により Trivial File Transfer Protocol (TFTP) を使用して外部ネットワークサーバーからロードされますが、すべての設定ファイルが **/flash** に配置されます。

また、起動スタックの上部にある起動ネットワークインターフェイスと起動ネットワーク設定のコマンドにも注意してください。これらのコマンドは、使用するリモート管理 LAN インターフェイスと、オペレーティングシステムソフトウェアのイメージファイルをホストする外部ネットワークサーバーとの通信に関する方法を定義します。

```
boot networkconfig static ip address mio1 192.168.1.150 netmask 255.255.255.0  
boot delay 15  
boot system priority 18 image tftp://192.168.1.161/tftpboot/image_version.bin \  
/flash/general_config.cfg  
boot system priority 19 image tftp://192.168.1.161/tftpboot/image_version.bin \  
/flash/general_config.cfg  
boot system priority 20 image /flash/image_version.bin \  
/flash/general_config.cfg
```

初期起動時にロードされた起動イメージの優先順位を確認するには、次のように入力します。

**show boot initial-config**

次に、出力例を示します。

```
[local]host_name# show boot initial-config  
Initial (boot time) configuration:  
image tftp://192.168.1.161/tftpboot/image_version.bin \  
config /flash/config_name.cfg  
priority 1
```

## 新しい起動スタックエントリの追加



**重要** この手順を実行する前に、`boot.sys` ファイルに 10 未満のエントリがあり、より高い優先順位のエントリが使用可能であることを確認します（つまり、少なくとも起動スタックに優先順位 1 のエントリがないことを確認します）。詳細については、「現在の起動スタックの表示」を参照してください。

優先順位 1 が使用されている場合は、既存のエントリの番号を付け直して、少なくともその優先順位が使用可能であることを確認する必要があります。`boot.sys` ファイルに含めることができる起動スタックエントリの最大数は 10 です。起動スタックにすでに 10 個のエントリがある場合は、これらのエントリのうち少なくとも 1 つを削除する必要があります（通常は優先順位が最も低いエントリ）。また、必要に応じて、他のエントリの一部またはすべての番号を再割り当てしてから続行します。詳細については、[起動スタックエントリの削除 \(146 ページ\)](#) を参照してください。

この手順では、新しい起動スタックエントリを `boot.sys` ファイルに追加する方法について説明します。Exec モードのプロンプトが表示されていることを確認し、次のコマンドを入力します。

**configure**

```
boot system priority number image image_url config cfg_url
```

次のコマンドは、起動優先順位 3 を使用して、新しい起動スタックエントリを作成します。

```
boot system priority 3 image /flash/image_filename.bin config
/flash/config_name.cfg
```




---

**重要** `boot.sys` ファイルに保存された起動スタックの変更は、システムが再起動されるまで実行されません。

---

次のコマンドを使用して、CF VM 上のローカルファイルシステムを同期します。

```
filesystem synchronize all
```

## 起動スタックエントリの削除

この手順では、`boot.sys` ファイルから個々の起動スタックエントリを削除する方法について説明します。Exec モードのプロンプトが表示されていることを確認し、次のコマンドを入力します。

**configure**

```
no boot system priority number
```

`number` は、起動スタックエントリに使用される起動の優先順位を指定します。このコマンドは、起動スタックから特定のエントリを削除して、`boot.sys` ファイルが上書きされるようにします。

## オペレーティングシステムソフトウェアのアップグレード

この項では、StarOS バイナリイメージのアップグレード手順を手動で実行する方法について説明します。

StarOS ソフトウェアのアップグレード手順を開始する前に、「前提条件」の項で説明されている条件が満たされていることを確認してください。




---

**注意** VPC の展開解除/再展開は、bin アップグレード後はサポートされません。VPC を非アクティブにすると、アップグレードした StarOS の bin イメージが削除されます。

---

StarOS ソフトウェアを手動でアップグレードするには、次の手順を実行します。

1. *AutoVNF*、*CF*、*ESC*、および *UEM* の *VIP* アドレスの取得
2. [OS リリースバージョンとビルド番号の識別 \(149 ページ\)](#)
3. [サポートサイトからソフトウェアイメージをダウンロード \(150 ページ\)](#)
4. *Zookeeper* データベースの確認
5. *ESC* データベースの確認
6. [/flash デバイスの空き領域の確認 \(152 ページ\)](#)
7. [StarOS イメージを /flash に転送 \(153 ページ\)](#)
8. [実行コンフィギュレーションの保存 \(158 ページ\)](#)
9. [ファイルシステムの同期 \(166 ページ\)](#)
10. [システムの再起動 \(160 ページ\)](#)

## 前提条件

CF と SF の VNFC を含む StarOS ソフトウェアのアップグレードを実行する前に、次の前提条件が満たされているかどうかを確認します。

- *AutoDeploy*、*AutoVNF*、*ESC*、*UEM*、および *CF VM* のログインクレデンシャルと IP アドレスが必要です。OpenStack 設定の管理権限を持っている必要があります。
- OpenStack のステータスを確認します。Ansible の出力がすべて引き渡される必要があります。

```
cd /home/stack/  
source stackrc  
cd /home/stack/ansible/  
ansible-playbook -i inventory openstack_verify.yml
```

- *AutoVNF/ESC/EM/VNF VM* の正常性が *AutoIT* の *UltraM* 正常性ログを通じて正常であるかどうかを確認します。いずれかの VM が正常でない場合は、対応する VM の正常性を回復するための必要なアクションを実行します。
- 新しい StarOS バイナリイメージファイル（手動アップグレードの場合）が必要です。
- *ESC*、*UEM*、および *CF* の間に保留中のトランザクションがないことを確認します。
- 元の StarOS bin ファイルのバックアップを必ず実行します。

## AutoVNF、CF、ESC、および UEM の VIP アドレスの取得

この項では、CF および SF VNFC のアップグレードにのみ適用される手順について説明します。

AutoVNF、CF、ESC、および UEM VM の VIP アドレスの収集

1. デフォルトのユーザーである *ubuntu* として AutoDeploy VM にログオンします。

```
ssh ubuntu@<ad_vm_address>
```

2. ルートユーザーに切り替えます。

```
sudo -i
```

3. ConfD CLI を入力します。

```
confd_cli -u admin -C
```

4. プロンプトが表示されたら、管理者ユーザーのログイン情報を入力します。

5. AutoVNF、ESC、UEM、および CF VM の VIP アドレスを収集します。

```
show vnfr
```

出力例：

```
vnfr autoit-f-autovnf
vnfd      f-autovnf
vnf-type  usp-uas
state     deployed
external-connection-point avf
virtual-link-ref      management
ip-address            192.168.100.26
floating-ip-address  10.225.202.94
vnfr sj-autovnf-esc
vnfd      esc
vnf-type  esc
state     deployed
external-connection-point esc
virtual-link-ref      management
ip-address            192.168.100.22
vnfr sj-autovnf-vpc
vnfd      vpc
vnf-type  ugp
state     alive
external-connection-point cf
virtual-link-ref      management
```

```
ip-address      192.168.100.38

external-connection-point em

virtual-link-ref management

ip-address      192.168.100.21
```

## OS リリースバージョンとビルド番号の識別

オペレーティングシステムは、CLIから発行されたコマンドを使用して、サービスを提供し、事前定義された機能を実行するように設定できます。

オペレーティングシステムのソフトウェアは、単一のバイナリファイル（ファイル拡張子 **.bin**）として提供され、システム全体の単一インスタンスとしてロードされます。

- イメージファイル名は、プラットフォームタイプとリリース番号を指定するサフィックスで識別されます。たとえば、**asr5500-release\_number.bin** というようになります。たとえば、**asr5500-16.1.0.bin** というようになります。

starfile イメージは、リリース前に REL キーで署名する必要があります。展開可能なイメージは、「.bin.SPA」拡張子を持つ REL キーを使用して署名されます。「A」は、署名キーのリビジョンレベルを示します。たとえば、**asr5500-20.0.0.bin.SPA** というようになります。署名キーが侵害されると、新しいキーが作成され、リビジョンレベルが「B」に増加します。

信頼できるイメージが導入されました。信頼できるビルドと通常のビルドの違いは、非セキュアなプログラムである `ftpd`、`telnet` および `tcpdump` がないことと、セキュリティオプション用の `staros.conf` ファイルが追加されていることです。信頼できるイメージは、プラットフォーム名に「\_T」が存在することによって識別できます。たとえば、**asr5500\_T-20.0.0.bin.SPA** というようになります。

StarOS ソフトウェアのバージョンとビルド情報を確認するには、次のようにします。

1. アップグレードする VNF にログオンします。
2. StarOS コマンドラインインターフェイスで次の Exec モードのコマンドを入力します。

```
show version
```

出力例：

```
Active Software:

Image Version:      21.9.0.69918

Image Build Number: 69918

Image Description:  Deployment_Build

Image Date:         Sun Jul 22 12:08:55 EDT 2018

Boot Image:         /flash/staros.bin

Source Commit ID:   94797337b6c1691541ea0dd86f2f29b0f2c3630c
```

3. StarOS ビルドリリースに関する追加情報を表示するには、次の Exec モードのコマンドを実行します。

```
show build
```

## サポートサイトからソフトウェアイメージをダウンロード

この項では、CF および SF VNFC のアップグレードにのみ適用される手順について説明します。

シスコのサポートサイトとダウンロードファシリティへのアクセスは、ユーザー名とパスワードで制御されています。サイトにアクセスして StarOS のイメージをダウンロードするには、アクティブなカスタマーアカウントが必要です。

*/flash* デバイスにアップロードできるネットワークの場所または物理デバイス（USB スティック）に、ソフトウェアイメージをダウンロードします。詳細についてはシスコの担当者または Cisco TAC にお問い合わせください。

UGP ベースの VNF の場合、次の手順を実行して、新しい bin ファイルを AutoVNF または OSPD VM にダウンロードします。

1. 対応する VNF の AutoVNF にログオンします。

```
ssh ubuntu@<ad_vm_address>
```

コマンドの例：

```
ssh ubuntu@10.225.202.94
```

2. 新しい StarOS qvpc-di バイナリファイルを AutoVNF/OSPD にダウンロードするためのディレクトリを作成します。

```
cd /home/ubuntu/
```

```
mkdir StarOSBinUpgrade
```

3. Cisco のサポートサイトから新しい StarOS qvpc-di バイナリファイルをダウンロードし、*StarOSBinUpgrade* ディレクトリにファイルをコピーします。

```
cd StarOSBinUpgrade
```

次のコマンドを使用して、ディレクトリに新しい bin ファイルが含まれているかどうかを確認します。

```
ls -lrt /home/ubuntu/StarOSBinUpgrade
```

出力例：

```
total 172560
```

```
-r--r--r-- 1 ubuntu ubuntu 176698880 Jul 24 23:29 qvpc-di-21.9.0.69932.bin
```

## Zookeeper データベースの確認

この項では、CF および SF VNFC のアップグレードにのみ適用される手順について説明します。

zookeeper データベースを確認するには、次のようにします。

1. フローティング IP を使用して AutoVNF にログオンします。

```
ssh ubuntu@<ad_vm_address>
```

コマンドの例：

```
ssh ubuntu@10.225.202.94
```

2. [AutoVNF、CF、ESC、および UEM の VIP アドレスの取得 \(147 ページ\)](#) で取得した VIP アドレスを使用して、UEM VM にログオンします。

```
ssh ubuntu@<vip-addr>
```

コマンドの例：

```
ssh ubuntu@192.168.100.21
```

3. ルートユーザーになります。

```
sudo -i
```

4. Zookeeper データベース接続の UEM オークストレーション IP アドレスを収集します。

```
#ifconfig
```

```
eth0      Link encap:Ethernet  HWaddr fa:16:3e:71:1d:08
```

```
          inet addr:209.165.200.240  Bcast: 209.165.200.255  Mask: 255.255.255.224
```

5. Navigate to the `/opt/cisco/usp/packages/zookeeper/<current>/bin` ディレクトリに移動します。
6. UEM Zookeeper データベースにアクセスするには、コマンドラインから次のスクリプトを実行します。

```
zkCli.sh -server ip_addr:port_num
```

次に例を示します。

```
zkCli.sh -server 209.165.200.240:2181
```

7. zookeeper データベースを確認し、UEM VM と CF VM の間に保留中の要求がないことを確認します。

```
ls /request
```

出力例：

```
[]
```

```
<Ctrl+D to exit Zookeeper shell>
```

## ESC データベースの確認

この項では、CF および SF VNFC のアップグレードにのみ適用される手順について説明します。

ESC データベースを確認するには、次のようにします。

1. フローティング IP を使用して AutoVNF にログオンします。

```
ssh ubuntu@<ad_vm_address>
```

コマンドの例：

```
ssh ubuntu@10.225.202.94
```

2. [AutoVNF、CF、ESC、および UEM の VIP アドレスの取得 \(147 ページ\)](#) で取得した VIP アドレスを使用して ESC VM にログオンします。

```
ssh admin@<vip-addr>
```

コマンドの例：

```
ssh admin@192.168.100.22
```

3. ESC データベースを確認して、保留中のトランザクションがないことを確認します。

```
sudo /opt/cisco/esc/pgsql/bin/psql -U esc -p 7878 -h localhost -c
'select * from esc_schema.workitem';
```

```
config_id | request_id | mo_type | config_action | config_state
-----+-----+-----+-----+-----
(0 rows)
```

4. トランザクションの詳細を確認するには、次のコマンドを実行します。

```
escadm ip_trans
```

出力例：

```
Number of in-progress transaction events = 0
```

## /flash デバイスの空き領域の確認

新しい StarOS イメージファイルに対応するために、/フラッシュデバイスに十分な空き領域があることを確認します。

フラッシュディレクトリで使用可能な領域を確認するには、次のようにします。

1. [AutoVNF、CF、ESC、および UEM の VIP アドレスの取得 \(147 ページ\)](#) で以前に取得した VIP アドレスを使用して CF VM にログインします。

```
ssh ubuntu@<vip-addr>
```

コマンドの例：

```
ssh ubuntu@192.168.100.38
```

2. 次の Exec モードコマンドを入力します。



```
[local]host_name# directory /flash
```

次に、表示されるディレクトリ情報のタイプの例を示します。

```
-rwxrwxr-x 1 root root 7334 May 5 17:29 asr-config.cfg
-rwxrwxr-x 1 root root 399 Jun 7 18:32 system.cfg
-rwxrwxr-x 1 root root 10667 May 14 16:24 testconfig.cfg
-rwxrwxr-x 1 root root 10667 Jun 1 11:21 testconfig_4.cfg
-rwxrwxr-x 1 root root 5926 Apr 7 16:27 tworpcontext.cfg
-rwxrwxr-x 1 root root 15534 Aug 4 13:31 test_vlan.cfg
-rwxrwxr-x 1 root root 2482 Nov 18 11:09 gateway2.cfg
-rwxrwxr-x 1 root root 159106048 Dec 31 2011 image_filename
1136352 /flash
Filesystem 1k-blocks Used Available Use% Mounted on
/var/run/storage/flash/part1 3115468 1136352 30018336 4%
/mnt/user/.auto/onboard/flash
```

ディスプレイの最後の行にある「Available」ブロックに注意してください。ディレクトリ情報を表示した後、CLI はルートに戻り、次のプロンプトが表示されます。

```
[local]host_name#
```

## StarOS イメージを /flash に転送

次のいずれかの方法を使用して、新しいオペレーティングシステムのイメージファイルを MIO/UMIO/MIO2 VPC-DI アクティブ CF または VPC-SI 上の /flash ディレクトリに転送します。

- システムへのアクセス権を持つ FTP クライアントを使用して、ファイルを /flash デバイスに転送します。



**重要** ファイル転送プロトコル (FTP) を使用してオペレーティングシステムのソフトウェアイメージファイルを転送する場合は、バイナリモードを使用してファイルを転送するように FTP クライアントを設定する必要があります。バイナリ転送モードを使用しないと、転送されたオペレーティングシステムイメージファイルが使用できなくなります。FTP はサポートされていません。

- システムへのアクセス権を持つ SFTP クライアントを使用して、ファイルを /flash デバイスに転送します。

UGP ベースの VNF の場合は、次の手順に従って、新しい StarOS bin をアクティブ CF にコピーします。

1. 新しい bin ファイルがダウンロードされた AutoVNF または OSPD VM にログオンします。

```
ssh ubuntu@<ad_vm_address>
```

コマンドの例：

```
ssh ubuntu@10.225.202.94
```

2. 新しい bin ファイルがシスコのサポートサイトからダウンロードされたディレクトリに移動します。

```
cd /home/ubuntu/StarOSBinUpgrade/ && ls -lrt
```

出力例：

```
total 172560
```

```
-r--r--r-- 1 ubuntu ubuntu 176698880 Jul 24 23:29 qvpc-di-21.9.0.69932.bin
```

3. SFTP から CF VM へ。

次に例を示します。

```
sftp ubuntu@192.168.100.38
```

4. *sftp* ディレクトリに移動します。

```
#sftp>pwd
```

```
Remote working directory: /
```

```
#sftp>ls
```

```
hd-raid sftp
```

```
#sftp>cd sftp
```

5. 新しいバイナリファイルを *sftp* ディレクトリにアップロードします。

```
#sftp>put image_filename.bin
```

出力例：

```
#sftp>put qvpc-di-21.9.0.69932.bin
```

```
Uploading qvpc-di-21.9.0.69932.bin to
```

```
/.auto/onboard/flash/sftp/qvpc-di-21.9.0.69932.bin
```

```
qvpc-di-21.9.0.69932.bin 100% 169MB 168.5MB/s 00:01
```

6. [AutoVNF、CF、ESC、およびUEMのVIPアドレスの取得 \(147ページ\)](#) で取得したVIPアドレスを使用してCF VMにログオンします。

```
ssh ubuntu@<vip-addr>
```

コマンドの例：

```
ssh ubuntu@192.168.100.38
```

7. 新しい bin を *sftp* から *flash* ディレクトリにコピーします。

```
copy /flash/sftp/image_filename.bin /flash/updated.bin
```

出力例：

```
#copy /flash/sftp/qvpc-di-21.9.0.69932.bin /flash/updated.bin
```

```
*****
```

```
Transferred 176698880 bytes in 2.718 seconds (63486.9 KB/sec)
```

8. *sftp* ディレクトリから新しい bin を削除します。

```
delete /flash/sftp/image_filename.bin
```

出力例：

```
delete /flash/sftp/qvpc-di-21.9.0.69932.bin
Are you sure? [Yes|No]: yes
File /flash/sftp/qvpc-di-21.9.0.69932.bin removed
```

9. 次の Exec モードのコマンドを実行して、イメージファイルが `/flash` デバイスに正常に転送されたことを確認します。

```
[local]host_name# directory /flash
```

表示された出力にイメージファイル名が表示されます。

10. 次のコマンドを実行して、ビルド情報を確認します。

```
show version /flash/image_filename.bin
```

## 現在の設定ファイルのコピーの保存

新しいソフトウェアリリースにアップグレードする前に、現在の設定ファイルを `/flash` デバイスとシャーシ外の場所（外部メモリデバイスやネットワーク URL）にコピーして名前を変更する必要があります。この名前が変更されたコピーが、アップグレード中に問題が発生した場合に使用できるフォールバック用のロードが可能な設定ファイルとなります。

## リリース 20.0 からのダウングレード

PBKDF2（パスワードベースのキー派生関数バージョン 2）を使用し、入力されたデータ、ソルト、および反復回数に基づいて、特定の長さのキーを取得するようになりました。ローカルユーザーアカウントのパスワードは、ランダムに生成されたソルトと多くの反復回数を備えた PBKDF2 方式を使用してハッシュされ、パスワードストレージの保護が強化されます。

MD5 ハッシュアルゴリズムを使用するようにローカルユーザーデータベースをダウングレードするには、セキュリティ管理者が Exec モードの `downgrade local-user database` コマンドを実行する必要があります。StarOS は確認のプロンプトを表示し、パスワードを再入力するようセキュリティ管理者に要求します。ダウングレードコマンドを実行する前に、ユーザーは入力したパスワードで再認証されます。確認後、パスワードは適切な古いまたは弱い暗号化アルゴリズムを使用してハッシュされ、データベースに保存されて、以前のバージョンの StarOS がセキュリティ管理者を認証できるようになります。

ダウングレードプロセスでは、PBKDF2 でハッシュされたパスワードは MD5 形式に変換されません。ダウングレードプロセスでは、（`/flash` ディレクトリから）データベースが再度読み込まれ、古い形式でデータベースが再構築されてからディスクに書き込まれます。PBKDF2 でハッシュされたパスワードは MD5 ハッシュアルゴリズムに変換できず、以前の StarOS リリースでは PBKDF2 暗号化アルゴリズムを解析できないため、StarOS は PBKDF2 アルゴリズムを介して暗号化されたすべてのユーザーを一時停止します。MD5 アルゴリズム（「弱いハッシュ」フラグ）を介して暗号化されたユーザーは、ログイン情報を使用してログインを続行できます。システムが以前の StarOS リリースで起動した後、一時停止されたユーザーは `show local-user [verbose]` コマンドの出力で確認できます。

一時停止されたユーザーを再アクティブ化するには、セキュリティ管理者が次の手順を実行します。

- Exec モードの **password change local-user username** コマンドを使用して、一時停止されたユーザーの一時パスワードを設定します。
- コンフィギュレーションモードの **no suspend local-user username** コマンドを使用して、ユーザーの一時停止フラグをリセットします。

## オフラインソフトウェアのアップグレード

オフラインソフトウェアのアップグレードは、バージョン番号に関係なく、任意のバージョンのオペレーティングシステムのソフトウェアから、任意のバージョンにアップグレードするすべてのシステムに対して実行できます。このプロセスは、システムが現在のセッションをサポートしている間に多くのステップを実行できますが、このプロセスの最後のステップでは、実際にソフトウェアのアップグレードを適用するために再起動する必要がありますので、オフラインと見なされます。

この手順では、CLIセッションが確立されており、新しいオペレーティングシステムのイメージファイルをローカルファイルシステムに配置していることを前提としています。開始するには、Exec モードのプロンプトが表示されていることを確認してください。

```
[local]host_name#
```

オフラインソフトウェアのアップグレードを実行するには：

1. [新規コールポリシーの設定 \(156 ページ\)](#)
2. [Day バナーのメッセージの設定 \(157 ページ\)](#)
3. [現在の CLI コンフィギュレーション ファイルのバックアップ \(157 ページ\)](#)
4. [実行コンフィギュレーションの保存 \(158 ページ\)](#)
5. [新しい起動スタックエントリの作成 \(159 ページ\)](#)
6. [ファイルシステムの同期 \(166 ページ\)](#)
7. [システムの再起動 \(160 ページ\)](#)

### 新規コールポリシーの設定

サービス要件を満たすために、Exec モードから新規コールポリシーを設定します。このポリシーを有効にすると、アップグレードプロセスが完了したシステムのリロードを予測して、新規コールがリダイレクトまたは拒否されます。これにより、アップグレードが完了したシステムのリロードによって発生するサブスクライバへのサービスの中断時間が短縮されます。



**重要** 新規コールポリシーは、サービス単位で作成されます。シャーシで複数のサービスを実行している場合は、複数の新規コールポリシーを設定できます。

新規コールポリシーのシンタックスについては、以下を参照してください。

```
[local]host_name# newcall policy { asngw-service | asnpc-service |
sgsn-service } { all | name service_name } reject
[local]host_name# newcall policy { fa-service | lns-service |
mipv6ha-service } { all | name service_name } reject
[local]host_name# newcall policy { ha-service | pdsn-service |
pdsnclosedrp-service } { all | name service_name } { redirect
target_ip_address [ weight weight_num ] [ target_ipaddress2 [ weight weight_num ]
... target_ip_address16 [ weight weight_num ] | reject }
[local]host_name# newcall policy ggsn-service { apn name apn_name | all |
name service_name } reject
[local]host_name# newcall policy hnbgw-service { all | name service_name }
reject
[local]host_name# newcall policy { pcc-af-service | pcc-policy-service } {
all | name service_name } reject
[local]host_name# newcall policy {pcc-af-service | pcc-policy-service } {
all | name service_name } reject
[local]host_name# newcall policy mme-service { all | name service_name }
reject
```

上記のコマンドの詳細については、『*Command Line Interface Reference*』の「*Exec Mode Commands*」の章を参照してください。

## Day バナーのメッセージの設定

オプション：グローバル コンフィギュレーション モードのプロンプトから次のコマンドを入力して、システムを再起動されることを他の管理ユーザーに通知する「Message of the Day」バナーを設定します。

```
[local]host_name(config)# banner motd "banner_text"
```

*banner\_text* は、表示するメッセージで、最大 2048 文字の英数字を使用できます。*Banner\_text* は、引用符で始める引用符で終わる ("") 必要があることに注意してください。CLI バナー情報の入力の詳細については、『*CLI Reference*』を参照してください。バナーは、管理ユーザーが CLI にログインしたときに表示されます。

## 現在の CLI コンフィギュレーション ファイルのバックアップ

次のコマンドを入力して、現在の CLI コンフィギュレーション ファイルをバックアップします。

```
[local]host_name# copy from_url to_url [ -noconfirm ]
```

これにより、現在の起動スタックエントリで定義されているオペレーティングシステムにリンクされている CLI コンフィギュレーション ファイルのミラーイメージが作成されます。

次のコマンド例では、*flash* デバイスにある *general.cfg* というファイルのバックアップコピーを、*general\_3652.cfg* というファイルに作成します。

```
[local]host_name# copy /flash/general.cfg /flash/general_3652.cfg
```

## 実行コンフィギュレーションの保存

シャーシを再起動する前に、現在実行中のアップグレード済みの設定を保存します。

起動設定を保存するには、次を実行します。

1. [AutoVNF、CF、ESC、およびUEMのVIPアドレスの取得 \(147 ページ\)](#) で以前に取得したVIPアドレスを使用してVNFにログオンします。

```
ssh ubuntu@<vip-addr>
```

コマンドの例：

```
ssh ubuntu@192.168.100.38
```

2. オプションです。Exec モードで次のコマンドを実行します。

```
chassis key value 1234
```

```
Save config before reload chassis, EVEN IF the same old key value is used.
Old config scripts will become invalid after reload.
```




---

**重要** この手順は任意であり、シャーシキーが設定されていない場合にのみ必要になります。

---

3. フラッシュディレクトリに起動設定を保存します。

```
save configuration /flash/system.cfg
```

```
Warning: About to overwrite boot configuration file
Are you sure? [Yes|No]: yes
```

これにより、新しい bin イメージを使用するように起動設定が更新されます。

次のコマンドを使用して起動設定を確認します。

```
# show boot
```

```
Monday May 21 20:39:57 UTC 2018
```

```
boot system priority 8 \
  image /flash/sftp/production.YYYYY.qvpc-di.bin \
  config /flash/sftp/tb5_vnf1_dayN.cfg
```

```
boot system priority 9 \
  image /flash/staros.bin \
  config /flash/sftp/tb5_vnf1_dayN.cfg
```

```
boot system priority 10 \
  image /flash/staros.bin \
  config /flash/system.cfg
```

4. コンフィギュレーションモードを開始し、新しい StarOS bin ファイルの起動優先順位を変更します。

```
#config
```

```
#boot system priority 1 image /flash/updated.bin config
/flash/system.cfg
```

```
#end
```

5. 新しい起動優先順位を確認します。

#### #show boot

```
boot system priority 1 \
    image /flash/updated.bin \
    config /flash/system.cfg
boot system priority 10 \
    image /flash/staros.bin \
    config /flash/system.cfg
```

6. フラッシュディレクトリに起動設定と新しい bin が含まれているかどうかを確認します。

#### dir /flash

```
total 320376
-rw-rw-r-- 1 root root 134 May 3 10:11 boot.sys
-rw-rw-r-- 1 root root 3920672 May 11 19:49 crashlog2
drwxrwxr-x 2 root root 4096 May 11 19:49 crsh2
-rw-rw-r-- 1 root root 156 May 11 19:49 module.sys
drwxrwxr-x 3 root root 4096 May 11 19:49 patch
drwxrwxr-x 2 root root 4096 May 11 19:49 persistdump
-rw-rw-r-- 1 root root 79 May 11 19:49 restart_file_cntr.txt
drwxrwxr-x 3 root root 4096 May 11 20:07 sftp
-rw-rw-r-- 1 root root 160871936 May 3 10:11 staros.bin
-rw-rw-r-- 1 root root 5199 May 11 19:57 system.cfg
-rw-rw-r-- 1 root root 163227136 May 11 20:07 updated.bin
320476 /flash
Filesystem 1K-blocks Used Available Use% Mounted on
/var/run/storage/boot1/part2
4112620 320476 3792144 8% /mnt/user/.auto/onboard/flash
```

## 新しい起動スタックエントリの作成

次のグローバルコンフィギュレーションコマンドを入力して、新しいオペレーティングシステムイのメージファイルと現在使用されているCLI設定ファイルで構成される新しいファイルグループの新しい起動スタックエントリを作成します。

```
[local]host_name(config)# boot system priority number image image_url /flash
filename config cfg_url /flash/filename
```

<N-1>方式を使用して、このエントリに次に高い優先順位を割り当てます。この場合、優先順位番号は、現在の最高の優先順位よりも1つ小さい値を割り当てます。




---

**重要** Exec モードの **show boot** コマンドを実行して、**boot.sys** ファイルに 10 未満のエントリがあることと、高い優先順位のエントリが使用可能である（最小で起動スタックに優先順位 1 のエントリがない）ことを確認します。

---

優先順位 1 が使用されている場合は、既存のエントリの数字を再割り当てし、少なくともその優先順位 1 を使用できるようにします。

boot.sys ファイルに含めることができる起動スタックエントリの最大数は10です。起動スタックにすでに10個のエントリがある場合は、これらのエントリのうち少なくとも1つを削除する必要があります（通常は優先順位が最も低いエントリ）。また、必要に応じて、他のエントリの一部またはすべての番号を再割り当てしてから続行します。起動スタックエントリを削除するには、`no boot system priority` コマンドを使用します。

```
[local]host_name# configure
[local]host_name(config)# no boot system priority number
```

新しい起動スタックエントリを boot.sys.sys ファイルに追加するには、次のコマンドを入力します。

```
[local]host_name# configure
[local]host_name(config)# boot system priority number image image_url config
cfg_url
```

`boot system priority` コマンドの使用方法については、[新しい起動スタックエントリの追加（145 ページ）](#) を参照してください。

## システムの再起動

システム（VNF）を再起動するには、次のようにします。

1. [AutoVNF、CF、ESC、および UEM の VIP アドレスの取得（147 ページ）](#) で以前に取得した VIP アドレスを使用して VNF にログオンします。

```
ssh ubuntu@<vip-addr>
```

コマンドの例：

```
ssh ubuntu@192.168.100.38
```

2. 次の Exec モードコマンドを入力します。

```
[local]host_name# reload [-noconfirm]
```

システムが再起動すると、以前に設定した新しい起動スタックエントリを使用して、新しいオペレーティングシステムのソフトウェアイメージとそれに対応する CLI コンフィギュレーションファイルがロードされます。

3. *PDSN* のオプション：アップグレード中に IP プール共有プロトコルを使用している場合は、『*PDSN Administration Guide*』の「*Configuring IPSP Before the Software Upgrade*」を参照してください。
4. リロードが完了したら、VNF にログオンして、目的の StarOS バージョンでロードされていること、およびすべてのカードが起動していること、および予想どおりにアクティブ状態またはスタンバイ状態になっていることを確認します。

```
show version
```

出力例：

```
Active Software:
Image Version:          21.9.0.69977
Image Build Number:     69977
Image Description:      Build
Image Date:             Mon Jul 30 06:48:34 EDT 2018
```



```

Boot Image: /flash/updated.bin
Source Commit ID: abde005a31c93734c89444b8aec2b6bb2d2e794d

```

### show card table

出力例 :

Slot	Card Type	Oper State	SPOF	Attach
1: CFC	Control Function Virtual Card	Active	No	
2: CFC	Control Function Virtual Card	Standby	-	
3: FC	4-Port Service Function Virtual Card	Standby	-	
4: FC	4-Port Service Function Virtual Card	Standby	-	
5: FC	4-Port Service Function Virtual Card	Standby	-	
6: FC	4-Port Service Function Virtual Card	Standby	-	
7: FC	4-Port Service Function Virtual Card	Standby	-	
8: FC	4-Port Service Function Virtual Card	Standby	-	
9: FC	4-Port Service Function Virtual Card	Standby	-	
10: FC	4-Port Service Function Virtual Card	Standby	-	

5. 次の Exec モードコマンドを実行して、実行中の StarOS ビルドリリースに関する追加情報を表示します。

### show build

6. オプションです。CF および SF VNFC の動作状態を確認します。



(注) このステップは、CF および SF VNFC をアップグレードする場合にのみ該当します。

1. 「Zookeeperデータベースの確認」と「ESCデータベースの確認」の項の手順を繰り返します。
2. フローティング IP を使用して UEM にログオンするか、または UEM VIP を使用して AutoVNF から UEM にログオンします。

```
ssh ubuntu@<vip-addr>
```

コマンドの例 :

```
ssh ubuntu@192.168.100.21
```

3. ルートユーザーになります。

```
sudo -i
```
4. Zookeeper データベース接続の UEM オーケストレーション IP アドレスを収集します。

```
#ifconfig
```

```
eth0      Link encap:Ethernet  HWaddr fa:16:3e:71:1d:08
```

```
          inet addr:209.165.200.225  Bcast:209.165.200.255  Mask:255.255.255.224
```
5. Navigate to the `/opt/cisco/usp/packages/zookeeper/<current>/bin` ディレクトリに移動します。
6. UEM Zookeeper データベースにアクセスするには、Zookeeper ツールを実行します。

```
zkCli.sh -server <vip-addr>:port_num
```

コマンドの例：

```
zkCli.sh -server 209.165.200.225:2181
```

UEM と CF の間に未処理の要求がないことを確認してください。

7. 次のコマンドを使用して、各 CF および SF の「状態」：「alive」を確認します。

```
get /oper/vnfs/vnf_name/vdus/vdu_name/cf1
```

```
get /oper/vnfs/vnf_name/vdus/vdu_name/cf2
```

```
get /oper/vnfs/vnf_name/vdus/vdu_name/sf1
```

```
get /oper/vnfs/vnf_name/vdus/vdu_name/sf2
```

コマンドの例：

```
get /oper/vnfs/tb1-autovnf1_vpc-vpc-core/vdus/vdu-cf1/cf1
```

```
get /oper/vnfs/tb1-autovnf1_vpc-vpc-core/vdus/vdu-cf1/cf2
```

```
get /oper/vnfs/tb1-autovnf1_vpc-vpc-core/vdus/vdu-sf1/sf1
```

```
get /oper/vnfs/tb1-autovnf1_vpc-vpc-core/vdus/vdu-sf1/sf2
```

8. コンソール出力で Alive 状態を確認します。

```
zk: localhost:2181(CONNECTED) 2] get
/oper/vdus/control-function/BOOT_generic_di-chassis_CF1_1
{"id":"BOOT_generic_di-chassis_CF1_1","state":"alive","vnfcId":"cf-vnfc-di-chassis","uuid":"c4",
"host":"tb5-ultram-osd-compute-2.localdomain","vimId":"523b921c-7266-4fd5-90bb-5157cffc6951",
"cpts":[{"cpid":"di_intf1","state":"alive","subnet":"6102e9b5-8555-41f5-8cdc-0b47d30a6f7a",
"netmask":"255.255.255.0","dhcp":true,"vl":"vl-vnf1-DI-INTERNAL1-CAT","vnfc":"cf-vnfc-di-chassis",
"port_id":"19539aea-edbf-4acf-a57c-af5627d859ea","ip_address":"192.168.10.3",
"mac_address":"fa:16:3e:19:80:ed","network":"0d72f553-5a9c-4904-b3ea-83371a806e23"},
{"cpid":"di_intf2","state":"alive","nicid":1,"subnet":"30002d02-761d-4ccb-8a9e-d6188cdf54a3",
"netmask":"255.255.255.0","dhcp":true,"vl":"vl-vnf1-DI-INTERNAL2-CAT","vnfc":"cf-vnfc-di-chassis",
"port_id":"ff1dale1-ecf3-477d-98b7-398c3c77fc8d","ip_address":"192.168.11.13",
"mac_address":"fa:16:3e:89:88:23","network":"9f109c0a-b1e7-4d90-a746-5de4ab8ef536"},
{"cpid":"orch","state":"alive","nicid":2,"subnet":"729e9dd2-3c75-43eb-988a-769016f2f44c",
"netmask":"255.255.255.0","dhcp":true,"vl":"vl-vnf1-UAS-ORCH-CAT","vnfc":"cf-vnfc-di-chassis",
"port_id":"81370948-f686-4812-820c-20ec5d3d3cdd","ip_address":"172.168.11.17","mac_address":"fa:16:3e:1d:0b:56",
"network":"9a286170-e393-4ba5-abce-147a45fb337a"}, {"cpid":"mgmt","state":"alive","nicid":3,
"subnet":"9778a11b-1714-4e84-bbc2-86c84b11e8e","netmask":"255.255.255.0","dhcp":true,"vl":"vl-vnf1-UAS-MGMT-CAT",
"vnfc":"cf-vnfc-di-chassis","port_id":"6130cbb4-3dd8-4822-af90-50dac98f2f0d",
"ip_address":"172.168.10.17","mac_address":"fa:16:3e:42:92:47","network":"e278b524-e9a9-48c1-a45b-956a8c3ea583"}],
"monitor":true,"vduId":"control-function"}
cZxid = 0x100000051
ctime = Fri May 18 19:04:40 UTC 2018
mZxid = 0x10000024a
mtime = Mon May 21 17:48:19 UTC 2018
pZxid = 0x100000051
cversion = 0
dataVersion = 12
aclVersion = 0
ephemeralOwner = 0x0
dataLength = 1625
numChildren = 0
```



(注) **CTRL+D** を使用して、zookeeper CLI を終了することができます。

9. UEM VM からルートユーザーとして、`ncs_cli` にログオンし、デバイスのライブステータスを確認します。

```
~$ sudo -i
```

```
ncs_cli -C -u admin
```

```
# show devices device device_name live-status
```

コマンド出力に、各カードの正しい「状態」と「カード状態」が反映されていることを確認します。

出力例：

```
# show devices device tb1-autovnfl_vpc-vpc-core-cf-nc live-status
```

```
<snip>
```

VNFC の参照	現在の状態	VNFC インスタンス ID	VDU の参照	カードタイプ	カード ID	コア数	CPU 使用率	ディスク容量	開始時刻	稼働時間	NOVA 起動	ID	日時	状態から	状態へ
cf1	-	cf1	cf	admin	1	-	-	-	-	-	-	-	-	-	-
cf2	-	cf2	cf	admin	2	-	-	-	-	-	-	-	-	-	-
sf1	-	sf1	sf	admin	3	-	-	-	-	-	-	-	-	-	-
sf2	-	sf2	sf	admin	4	-	-	-	-	-	-	-	-	-	-

```
live-status vnfd sj-autovnf-vpc-abc
```

```
version 6.0
```

```
vnfm vim-tenant-name abc
```

```
vnfm tenant-name abc
```

```
vnfm ipaddr 192.168.100.22
```

```
vnfm port 830
```

```
vnfm username ubuntu
```

```
vnfm password "$4$+HLzhFFzHq66nqtTsc00CfiODYHqlUSVmknltRelf84byNakWEa9sJ8sY/cwfFME3aG0UaBC\nvvNNAMkuXQI9Ks fu5IiQQ9ViWbbHw16IEFQ="
```

```
virtual-link vl-di-internall1
```

```
auto-vnf-connection-ref di-internall1
```

```
virtual-link vl-management
```

```

    auto-vnf-connection-ref management
virtual-link vl-orchestration
    auto-vnf-connection-ref orchestration
virtual-link vl-abc-vpc-svc
    auto-vnf-connection-ref sj-autovnf-abc-vpc-svc
vdu cf
    ssh-keygen          false
    vm-image            076c887a-a12c-4a0b-b4d6-b2d213f64b9e
    lifecycle-event-initialization staros_config.txt
    source-url
http://192.168.100.9:5000/config/sj-autovnf-vpc-abc/cf/staros_config.txt
    lifecycle-event-initialization staros_param.cfg
    source-url
http://192.168.100.9:5000/config/sj-autovnf-vpc-abc/cf/staros_param.cfg
    ned cisco-staros-nc
    user                "$4$+HLzsElkLJOeufWyoSmBWY2LHjOi2WtJdKy/OIux7YHhsNY/
08hnA9/WwWuFD5trHrW3ZHs\nLo4TfiAKqYwxdNKqFYyoTxH2hrLJV5DgwmE="
    password
"$4$+HLzsXtCHJ2vsYZD5s0RGtBRY/dHDU1mgHJX7wCt3o1DMtQZqpBLDcNSJumC7n5rnkVxwI1s\
ncJYeCOFLrqpLHXm3xtXyMdtT7WVzvRMtdao="
    netconf
    port-number 830
    card-type          control-function
    usp-auto-vnf-id   sj-autovnf-vpc-abc-cf
    vnf cf-vnfc-ugp
<snip>

```

## 実行コンフィギュレーションの保存

シャーシを再起動する前に、現在実行中のアップグレード済みの設定を保存します。

起動設定を保存するには、次を実行します。

1. [AutoVNF、CF、ESC、およびUEMのVIPアドレスの取得（147ページ）](#) で以前に取得したVIPアドレスを使用してVNFにログオンします。

```
ssh ubuntu@<vip-addr>
```

コマンドの例：

```
ssh ubuntu@192.168.100.38
```

- オプションです。Exec モードで次のコマンドを実行します。

```
chassis key value 1234
```

```
Save config before reload chassis, EVEN IF the same old key value is used.  
Old config scripts will become invalid after reload.
```



**重要** この手順は任意であり、シャーンキーが設定されていない場合にのみ必要になります。

- フラッシュディレクトリに起動設定を保存します。

```
save configuration /flash/system.cfg
```

```
Warning: About to overwrite boot configuration file  
Are you sure? [Yes|No]: yes
```

これにより、新しい bin イメージを使用するように起動設定が更新されます。

次のコマンドを使用して起動設定を確認します。

```
# show boot
```

```
Monday May 21 20:39:57 UTC 2018
```

```
boot system priority 8 \  
  image /flash/sftp/production.YYYYY.qvpc-di.bin \  
  config /flash/sftp/tb5_vnfl_dayN.cfg
```

```
boot system priority 9 \  
  image /flash/staros.bin \  
  config /flash/sftp/tb5_vnfl_dayN.cfg
```

```
boot system priority 10 \  
  image /flash/staros.bin \  
  config /flash/system.cfg
```

- コンフィギュレーションモードを開始し、新しい StarOS bin ファイルの起動優先順位を変更します。

```
#config
```

```
#boot system priority 1 image /flash/updated.bin config  
/flash/system.cfg
```

```
#end
```

- 新しい起動優先順位を確認します。

```
#show boot
```

```
boot system priority 1 \  
  image /flash/updated.bin \  
  config /flash/system.cfg  
  
boot system priority 10 \  
  image /flash/staros.bin \  
  config /flash/system.cfg
```

```
image /flash/staros.bin \
config /flash/system.cfg
```

6. フラッシュディレクトリに起動設定と新しい bin が含まれているかどうかを確認します。

#### dir /flash

```
total 320376
-rw-rw-r-- 1 root root 134 May 3 10:11 boot.sys
-rw-rw-r-- 1 root root 3920672 May 11 19:49 crashlog2
drwxrwxr-x 2 root root 4096 May 11 19:49 crsh2
-rw-rw-r-- 1 root root 156 May 11 19:49 module.sys
drwxrwxr-x 3 root root 4096 May 11 19:49 patch
drwxrwxr-x 2 root root 4096 May 11 19:49 persistdump
-rw-rw-r-- 1 root root 79 May 11 19:49 restart_file_cntr.txt
drwxrwxr-x 3 root root 4096 May 11 20:07 sftp
-rw-rw-r-- 1 root root 160871936 May 3 10:11 staros.bin
-rw-rw-r-- 1 root root 5199 May 11 19:57 system.cfg
-rw-rw-r-- 1 root root 163227136 May 11 20:07 updated.bin
320476 /flash
Filesystem 1K-blocks Used Available Use% Mounted on
/var/run/storage/boot1/part2
4112620 320476 3792144 8% /mnt/user/.auto/onboard/flash
```

## ファイルシステムの同期

ファイルシステムを同期するには、次の手順を実行します。

1. [AutoVNF、CF、ESC、および UEM の VIP アドレスの取得 \(147 ページ\)](#) で取得した VIP アドレスを使用して VNF にログオンします。
2. 次のコマンドを入力して、管理カード上のローカルファイルシステムを同期します。

```
[local]host_name# filesystem synchronize all
```

出力例 :

```
Updating /flash/system.cfg
```

```
*****
```

```
Updating /flash/updated.bin
```

```
*****
```

```
Updating /flash/sftp/yang/cisco-staros-bulkstats-config.yang
```

```
*****
```

```
Updating /flash/sftp/yang/cisco-staros-bulkstats-schema-types.yang
```

```
*****
```

```
Updating /flash/sftp/yang/cisco-staros-bulkstats.yang
```

```
*****
```

```
Updating /flash/sftp/yang/cisco-staros-cli-config.yang
```

```
*****
```

```

Updating /flash/sftp/yang/cisco-staros-confd-config.yang
*****
Updating /flash/sftp/yang/cisco-staros-config.yang
*****
Updating /flash/sftp/yang/cisco-staros-exec.yang
*****
Updating /flash/sftp/yang/cisco-staros-kpi.yang
*****
Updating /flash/sftp/yang/cisco-staros-notif.yang
*****
Updating /flash/boot.sys
*****
12 updated on card 2

    /flash/system.cfg
    /flash/updated.bin
    /flash/sftp/yang/cisco-staros-bulkstats-config.yang
    /flash/sftp/yang/cisco-staros-bulkstats-schema-types.yang
    /flash/sftp/yang/cisco-staros-bulkstats.yang
    /flash/sftp/yang/cisco-staros-cli-config.yang
    /flash/sftp/yang/cisco-staros-confd-config.yang
    /flash/sftp/yang/cisco-staros-config.yang
    /flash/sftp/yang/cisco-staros-exec.yang
    /flash/sftp/yang/cisco-staros-kpi.yang
    /flash/sftp/yang/cisco-staros-notif.yang
    /flash/boot.sys

```

## システムの再起動

システム（VNF）を再起動するには、次のようにします。

1. [AutoVNF、CF、ESC、および UEM の VIP アドレスの取得（147 ページ）](#) で以前に取得した VIP アドレスを使用して VNF にログオンします。

```
ssh ubuntu@<vip-addr>
```

コマンドの例：

```
ssh ubuntu@192.168.100.38
```

2. 次の Exec モードコマンドを入力します。

```
[local]host_name# reload [-noconfirm]
```

システムが再起動すると、以前に設定した新しい起動スタックエントリを使用して、新しいオペレーティングシステムのソフトウェアイメージとそれに対応する CLI コンフィギュレーションファイルがロードされます。

3. *PDSN* のオプション：アップグレード中に IP プール共有プロトコルを使用している場合は、『*PDSN Administration Guide*』の「*Configuring IPSP Before the Software Upgrade*」を参照してください。
4. リロードが完了したら、VNF にログオンして、目的の StarOS バージョンでロードされていること、およびすべてのカードが起動していること、および予想どおりにアクティブ状態またはスタンバイ状態になっていることを確認します。

```
show version
```

出力例：

```
Active Software:
  Image Version:          21.9.0.69977
  Image Build Number:    69977
  Image Description:     Build
  Image Date:            Mon Jul 30 06:48:34 EDT 2018
  Boot Image:            /flash/updated.bin
  Source Commit ID:     abde005a31c93734c89444b8aec2b6bb2d2e794d
```

```
show card table
```

出力例：

Slot	Card Type	Oper State	SPOF	Attach
1: CFC	Control Function Virtual Card	Active	No	
2: CFC	Control Function Virtual Card	Standby	-	
3: FC	4-Port Service Function Virtual Card	Standby	-	
4: FC	4-Port Service Function Virtual Card	Standby	-	
5: FC	4-Port Service Function Virtual Card	Standby	-	
6: FC	4-Port Service Function Virtual Card	Standby	-	
7: FC	4-Port Service Function Virtual Card	Standby	-	
8: FC	4-Port Service Function Virtual Card	Standby	-	
9: FC	4-Port Service Function Virtual Card	Standby	-	
10: FC	4-Port Service Function Virtual Card	Standby	-	

5. 次の Exec モードコマンドを実行して、実行中の StarOS ビルドリリースに関する追加情報を表示します。

```
show build
```

6. オプションです。CF および SF VNFC の動作状態を確認します。



(注) このステップは、CF および SF VNFC をアップグレードする場合にのみ該当します。



1. 「Zookeeperデータベースの確認」と「ESCデータベースの確認」の項の手順を繰り返します。
2. フローティング IP を使用して UEM にログオンするか、または UEM VIP を使用して AutoVNF から UEM にログオンします。

```
ssh ubuntu@<vip-addr>
```

コマンドの例 :

```
ssh ubuntu@192.168.100.21
```

3. ルートユーザーになります。
4. Zookeeper データベース接続の UEM オーケストレーション IP アドレスを収集します。

```
#ifconfig
```

```
eth0      Link encap:Ethernet  HWaddr fa:16:3e:71:1d:08
```

```
          inet addr:209.165.200.225  Bcast:209.165.200.255  Mask:255.255.255.224
```

5. Navigate to the `/opt/cisco/usp/packages/zookeeper/<current>/bin` ディレクトリに移動します。
6. UEM Zookeeper データベースにアクセスするには、Zookeeper ツールを実行します。

```
zkCli.sh -server <vip-addr>:port_num
```

コマンドの例 :

```
zkCli.sh -server 209.165.200.225:2181
```

UEM と CF の間に未処理の要求がないことを確認してください。

7. 次のコマンドを使用して、各 CF および SF の「状態」 : 「alive」を確認します。

```
get /oper/vnfs/vnf_name/vdus/vdu_name/cf1
```

```
get /oper/vnfs/vnf_name/vdus/vdu_name/cf2
```

```
get /oper/vnfs/vnf_name/vdus/vdu_name/sf1
```

```
get /oper/vnfs/vnf_name/vdus/vdu_name/sf2
```

コマンドの例 :

```
get /oper/vnfs/tb1-autovnf1_vpc-vpc-core/vdus/vdu-cf1/cf1
```

```
get /oper/vnfs/tb1-autovnf1_vpc-vpc-core/vdus/vdu-cf1/cf2
```

```
get /oper/vnfs/tb1-autovnf1_vpc-vpc-core/vdus/vdu-sf1/sf1
```

```
get /oper/vnfs/tb1-autovnf1_vpc-vpc-core/vdus/vdu-sf1/sf2
```

8. コンソール出力で Alive 状態を確認します。

```
zk: localhost:2181(CONNECTED) 2] get
/oper/vdus/control-function/BOOT_generic_di-chassis_CF1_1
{"id":"BOOT_generic_di-chassis_CF1_1","state":"alive","vnfcId":"cf-vnfc-di-chassis","uuid":"c4",
```

```

"host":"tb5-ultram-osd-compute-2.localdomain","vimId":"523b921c-7266-4fd5-90bb-5157cffc6951",
"cppts":[{"cpid":"di_intf1","state":"alive","subnet":"6102e9b5-8555-41f5-8cdc-0b47d30a6f7a",
"netmask":"255.255.255.0","dhcp":true,"vl":"vl-vnfl-DI-INTERNAL1-CAT","vnfc":"cf-vnfc-di-chassis",
"port_id":"19539aea-edbf-4acf-a57c-af5627d859ea","ip_address":"192.168.10.3",
"mac_address":"fa:16:3e:19:80:ed","network":"0d72f553-5a9c-4904-b3ea-83371a806e23"}],
{"cpid":"di_intf2","state":"alive","nicid":1,"subnet":"30002d02-761d-4ccb-8a9e-d6188cdf54a3",
"netmask":"255.255.255.0","dhcp":true,"vl":"vl-vnfl-DI-INTERNAL2-CAT","vnfc":"cf-vnfc-di-chassis",
"port_id":"ff1dale1-ecf3-477d-98b7-398c3c77fc8d","ip_address":"192.168.11.13",
"mac_address":"fa:16:3e:89:88:23","network":"9f109c0a-b1e7-4d90-a746-5de4ab8ef536"}],
{"cpid":"orch","state":"alive","nicid":2,"subnet":"729e9dd2-3c75-43eb-988a-769016f2f44c",
"netmask":"255.255.255.0","dhcp":true,"vl":"vl-vnfl-UAS-ORCH-CAT","vnfc":"cf-vnfc-di-chassis",
"port_id":"81370948-f686-4812-820c-20ec5d3d30d1","ip_address":"172.168.11.17","mac_address":"fa:16:3e:1d:0b:56",
"network":"9a286170-e393-4ba5-abce-147a45fb337a"}],{"cpid":"mgmt","state":"alive","nicid":3,
"subnet":"9778a11b-1714-4e84-bbc2-86c84b11e8e","netmask":"255.255.255.0","dhcp":true,"vl":"vl-vnfl-UAS-MGMT-CAT",
"vnfc":"cf-vnfc-di-chassis","port_id":"6130cbb4-3dd8-4822-af90-50dac98f2f0d",
"ip_address":"172.168.10.17","mac_address":"fa:16:3e:42:92:47","network":"e278b524-e9a9-48c1-a45b-956a8c3ea583"}],
"monitor":true,"vduId":"control-function"}
cZxid = 0x100000051
ctime = Fri May 18 19:04:40 UTC 2018
mZxid = 0x10000024a
mtime = Mon May 21 17:48:19 UTC 2018
pZxid = 0x100000051
cversion = 0
dataVersion = 12
aclVersion = 0
ephemeralOwner = 0x0
dataLength = 1625
numChildren = 0

```



(注) CTRL+D を使用して、zookeeper CLI を終了することができます。

- UEM VM からルートユーザーとして、`ncs_cli` にログオンし、デバイスのライブステータスを確認します。

```
~$ sudo -i
```

```
ncs_cli -C -u admin
```

```
# show devices device device_name live-status
```

コマンド出力に、各カードの正しい「状態」と「カード状態」が反映されていることを確認します。

出力例：

```
# show devices device tb1-autovnfl_vpc-vpc-core-cf-nc live-status
```

```
<snip>
```

VNFC	現在の状態	VNFC の状態	VDU の状態	カードタイプ	カード ID	コア数	CPU 使用率	ディスク容量	開始時刻	稼働時間	NOVA 起動	ID	日時	状態	状態
照	態	照	照	照	照	照	照	照	照	照	照	照	照	照	照
cf1	-	cf1	cf	active	1	-	-	-	-	-	-	-	-	-	-
cf2	-	cf2	cf	active	2	-	-	-	-	-	-	-	-	-	-

```

sf1 - sf1 sf staros 3 - - - - -
sf2 - sf2 sf staros 4 - - - - -

live-status vnfd sj-autovnf-vpc-abc
version          6.0
vnfm vim-tenant-name abc
vnfm tenant-name abc
vnfm ipaddr      192.168.100.22
vnfm port        830
vnfm username    ubuntu
vnfm password    "$4$+HLzhFFzHq66nqtTsc00CfiODYHqlUSVmkn1tRelf84byNakWEa9sJ8sY/
cwffME3aG0UaBC\nvvNNAMkuXQI9Ksfu5TiQQ9ViWbbHw16IEFQ="

virtual-link vl-di-internall
  auto-vnf-connection-ref di-internall

virtual-link vl-management
  auto-vnf-connection-ref management

virtual-link vl-orchestration
  auto-vnf-connection-ref orchestration

virtual-link vl-abc-vpc-svc
  auto-vnf-connection-ref sj-autovnf-abc-vpc-svc

vdu cf

ssh-keygen        false

vm-image          076c887a-a12c-4a0b-b4d6-b2d213f64b9e

lifecycle-event-initialization staros_config.txt

  source-url
http://192.168.100.9:5000/config/sj-autovnf-vpc-abc/cf/staros_config.txt

  lifecycle-event-initialization staros_param.cfg

  source-url
http://192.168.100.9:5000/config/sj-autovnf-vpc-abc/cf/staros_param.cfg

ned cisco-staros-nc

  user            "$4$+HLzsElkLJOeufWyoSmBWy2LHjOi2WtJdKy/OIux7YHhsNY/
O8hnA9/WwWuFD5trHrW3ZHS\nLo4TfiAKqYwxdNKqFYyoTxH2hrLJV5DgwmE="

  password
"$4$+HLzsXtCHJ2vsYZD5s0RGtBRY/dHDU1mgHJX7wCt3o1DMtQZqpBLDcNsJumC7n5rnkVxwI1s\
ncJYeCOFLrqpLHXm3xtXyMdtT7WVzvRMtdao="

```

```

netconf

port-number 830

card-type          control-function

usp-auto-vnf-id   sj-autovnf-vpc-abc-cf

vnfc cf-vnfc-ugp

<snip>

```

## 以前のソフトウェアイメージの復元

何らかの理由でアップグレードを元に戻す必要がある場合は、次の場合を除き、アップグレードを再度実行します。

- アップグレードソフトウェアイメージと設定ファイルの場所を指定します。

次に

- 元のソフトウェアイメージと設定ファイルの場所を指定します。

## ライセンスキーの管理

ライセンスキーは、キャパシティ制限（許可されるサブスクライバセッションの数）とシステムで使用可能な機能を定義します。新しいライセンスキーを追加すると、キャパシティを増やしたり、サブスクライバベースが増加したときに新しい機能を追加したりすることができます。

## 新しいシステムライセンスキー

新しいシステムは、ライセンスキーがインストールされていない状態で納品されます。ほとんどの場合、ライセンスキーは電子形式（通常は電子メール）で渡されます。

ライセンスキーがインストールされていない状態でシステムが起動すると、デフォルトの制限付きのセッション使用ライセンスと機能ライセンス一式がインストールされます。次の Exec モードコマンドは、ライセンス情報を一覧表示します。

```
[local]host_name# show license information
```




---

**重要** ライセンスキーがインストールされていない場合、PDSN、HA、GGSN、および L2TP LNS のセッション使用ライセンスは 10,000 セッションに制限されます。

---

## セッション使用とライセンス使用のライセンス

セッション使用および機能の使用ライセンスは、セッション制限を制御し、システム内の特別な機能を有効にするソフトウェアメカニズムです。これらの電子ライセンスは、システムが電源オンになるか再起動されるたびに、システムソフトウェアの一部としてロードされるシステムコンフィギュレーションファイルに保存されます。

- セッション使用ライセンスは、システムがサービスタイプごとにサポートできる同時セッション数を制限し、必要に応じて取得されます。これにより、キャリアは利用しているものに対してのみ支払いを行い、サブスクライバベースの増加に応じてキャパシティを簡単に増やすことができます。
- 機能使用ライセンスは、システム内で特定の機能を有効にし、システムでサポートされているセッションの合計数に基づいて配布されます。

## 新しいライセンスキーのインストール

新しいライセンスキーをインストールするには、次の手順を使用します。

### キーのカットアンドペースト

ライセンスのコピーがある場合は、次の設定を使用して、ライセンスキー部分だけをカットアンドペーストします。

**ステップ 1** Exec モードで、次のように入力します。

```
configure
license key license
exit
```

*license* はライセンスキー文字列です。ライセンスは、大文字と小文字が区別される 1 ~ 1023 文字の英数字文字列にすることができます。次の例に示すように、ライセンスキーをコピーします（「\」（二重引用符）を含む）。これは機能ライセンスではありませんのでご注意ください。

```
"\
VER=1|C1M=000-0000-00|C1S=03290231803|C2M=11-1111-11-1|C2S=\
STCB21M82003R80411A4|DOI=0000000000|DOE=00000000|ISS=1|NUM=13459|00000000000000|
LSP=000000|LSH=000000|LSG=500000|LSL=500000|FIS=Y|FR4=Y|FPP=Y|FCS=Y|FTC=Y|FMG=Y|
FCR=Y|FSR=Y|FPM=Y|FID=Y|SIG=MCwCF\Esnq6Bs/
XdmyfLe7rHcD4sVP2bzAhQ3IeHDoyyd6388jHsHD99sg36SG267gshssja77
end
```

**ステップ 2** Exec モードプロンプトで次のコマンドを入力して、入力したライセンスキーが受け入れられたことを確認します。

```
[local]host_name# show license key
```

新しいライセンスキーが表示されます。表示されない場合は、グローバルコンフィギュレーションモードに戻り、**license key** コマンドを使用してキーを再入力します。

## ライセンスキーをコンフィギュレーションファイルに追加

**重要** 無効なライセンスは受け入れられません。無効なライセンスキーを設定しようとする、**license key** コマンドの出力に障害エラーが表示されます。**-force** オプションを使用して無効なライセンスキーをインストールすると、ライセンスは30日間の猶予期間に入ります。StarOSは、猶予期間中に毎日のsyslogエラーメッセージとSNMPトラップを生成します。**show license information** コマンドの出力には、「License State」が「Not Valid」として示されます。

**ステップ3** 次のコマンドを入力して、ライセンスキーが正しい機能を有効にしていることを確認します。

```
[local]host_name# show license information
```

すべてのライセンスキーと、有効になっている新しいセッションのキャパシティまたは機能が表示されます。新しいキーで有効になっている機能またはセッションのキャパシティが正しくない場合は、サービス担当者にお問い合わせください。

**ステップ4** 「設定の確認と保存」の章の説明に従って、設定を保存します。

**注意** 新しいライセンスキー設定を現在のCLIコンフィギュレーションファイルに保存しないと、システムがリロードされた際に、ライセンスキーによって有効にされた新しい機能が失われます。

## ライセンスキーをコンフィギュレーションファイルに追加

ライセンスキーは、新規または既存のコンフィギュレーションファイルに追加できます。



**重要** ライセンスキー情報は、CLI設定の一部として維持されます。キーがインストールまたは更新されるたびに、コンフィギュレーションファイルを再保存する必要があります。

**ステップ1** 新しいライセンスキーコマンドをコピーするコンフィギュレーションファイルを開きます。

**ステップ2** 例に示すように、「\ (二重引用符)」を含むライセンスをコピーします。これは機能ライセンスではありませんのでご注意ください。

```
"\
VER=1|C1M=000-0000-00|C1S=03290231803|C2M=11-1111-11-1|C2S=\STCB21M82003R80411A4|
DOI=0000000000|DOE=00000000|ISS=1|NUM=13459|000000000000|LSP=000000|LSH=000000|
LSG=500000|LSL=500000|FIS=Y|FR4=Y|FPP=Y|FCS=Y|FTC=Y|FMG=Y|FCR=Y|FSR=Y|FPM=Y|FID=Y|
SIG=MCwCF\Esnq6Bs/XdmyfLe7rHcD4sVP2bzAhQ3IeHDoyyd6388jHsHD99sg36SG267gshssja77
end
```

**ステップ3** ライセンスキーを設定に貼り付けます。

**重要** コンフィギュレーションファイルの先頭にライセンスキー情報を貼り付けて、システムがコンテキストを設定する前に期待される容量と機能を確保するようにします。

**ステップ4** 「設定の確認と保存」の章の説明に従って、設定を保存します。

## ライセンスの期限切れの動作

ライセンスの有効期限が切れると、30日の猶予期間が設定され、ライセンス済みセッション使用および機能使用ライセンスの通常の使用が可能になります。これにより、サービスを中断せずに新しいライセンスを取得できます。

次の Exec モードのコマンドは、猶予期間の有効期限に設定された日付を含むライセンス情報を一覧表示します。

```
show license information
```

## ライセンスキーの要求

システムのライセンスキーは、シスコのアカウント担当者から入手できます。ライセンスキーを生成するには、特定の情報が必要です。

- SO または発注書情報
- 必要なセッションキャパシティ
- 必要な機能

## ライセンス情報の表示

ライセンスの詳細を表示するには、Exec モードで次のコマンドを入力します。

```
[local]host_name# show license information [ full | key [ full ] ]
```

## ライセンスキーの削除

セッションおよび機能使用ライセンスキーを設定から削除するには、次の手順を実行します。セキュリティ管理者または管理者である必要があります。

```
configure
  no license key
  exit
show license key
```

このコマンドの出力には、「No license key installed」と表示されています。

## ローカルユーザー管理アカウントの管理

設定ファイルを介して設定されたコンテキストレベルの管理アカウントとは異なり、ローカルユーザー管理アカウントの情報は、フラッシュメモリ内の別のファイルに保持され、ソフトウェアの共有設定タスク (SCT) によって管理されます。ローカルユーザーアカウントはANSI T1.276-2003 に準拠するように設計されているため、システムにはこれらのタイプの管理ユーザーアカウントを管理するためのさまざまなメカニズムが備わっています。

詳細については、[コンソールの AAA ベースの認証を無効化 \(92 ページ\)](#) および [コンソール/VTY 回線でのローカルユーザーログインの制限 \(93 ページ\)](#) を参照してください。

## ローカルユーザーパスワードのプロパティの設定

ローカルユーザーアカウントのパスワードプロパティはグローバルに設定され、すべてのローカルユーザーアカウントに適用されます。システムでは、次のパスワードプロパティの設定がサポートされています。

- **Complexity** : パスワードの複雑さは、ANSI T1.276-2003 に強制的に準拠させることができます。
- **History length** : システムで追跡する必要がある以前のパスワードバージョンの数。
- **Maximum age** : ユーザーが同じパスワードを使用できる期間。
- **Minimum number of characters to change** : リセット時にパスワードで変更する必要がある文字数。
- **Minimum change interval** : ユーザーがパスワードを変更できる頻度。
- **Minimum length** : 有効なパスワードに含める必要がある最小文字数。
- **Expiry warning** : パスワードの有効期限の警告間隔 (日数)。
- **Auto-generate** : パスワードの長さを指定するオプションを使用して、自動的にパスワードを生成します。

上記パラメータそれぞれの詳細については、『*Command Line Interface Reference*』の「*Global Configuration Mode Commands*」の章で説明されている **local-user password** コマンドを参照してください。

## ローカルユーザーのアカウント管理プロパティの設定

ローカルユーザーのアカウント管理には、アカウントのロックアウトとユーザーの一時停止が含まれています。

### ローカルユーザーアカウントのロックアウト

ローカルユーザーアカウントは、次の理由で管理上ロックすることができます。

- **ログインの失敗** : 設定された最大ログイン失敗のしきい値に達しました。詳細については、『*Command Line Interface Reference*』の「*Global Configuration Mode Commands*」の章で説明されている **local-user max-failed-logins** コマンドを参照してください。
- **パスワードのエイジング** : 設定された最大パスワードの有効期限に達しました。詳細については、『*Command Line Interface Reference*』の「*Global Configuration Mode Commands*」の章で説明されている **local-user password** コマンドを参照してください。



ロックアウトされたアカウントは、設定されたロックアウト時間に達するまで（『*Command Line Interface Reference*』の「*Global Configuration Mode Commands*」の章で説明されている **local-user lockout-time** コマンドを参照）、またはセキュリティ管理者がロックアウトをクリアするまで（『*Command Line Interface Reference*』の「*Exec Mode Commands*」の章で説明されている **clear local-user** コマンドを参照）、ユーザーはアクセスできません。



**重要** ローカルユーザーの管理ユーザーアカウントは、ロックアウトを適用または拒否するように設定できます。詳細については、『*Command Line Interface Reference*』の「*Global Configuration Mode Commands*」の章で説明されている **local-user username** コマンドを参照してください。

## ローカルユーザーアカウントの一時停止

ローカルユーザーアカウントは、次のように一時停止することができます。

```
configure
suspend local-user name
```

次のように入力して、一時停止を削除できます。

```
configure
no suspend local-user name
```

## ローカルユーザーパスワードの変更

ローカルユーザーの管理ユーザーは、Exec モードで **password change** コマンドを使用してパスワードを変更できます。ユーザーは、現在のパスワードと新しいパスワードを入力するように求められます。

セキュリティ管理者は、Exec モードで **root** プロンプトから次のコマンドを入力して、ローカルユーザーのパスワードをリセットできます。

```
[local]host_name# password change username name
```

**name** は、パスワードを変更するローカルユーザーアカウントの名前です。セキュリティ管理者がローカルユーザーのパスワードをリセットすると、ユーザーは次回ログイン時にパスワードを変更するように求められます。

新しいパスワードは、システムに設定されているパスワードプロパティに従う必要があります。





## 第 12 章

# スマートライセンス

- 機能の概要と変更履歴 (179 ページ)
- スマート ソフトウェア ライセンシング (180 ページ)
- スマート ライセンスの設定 (185 ページ)
- スマートライセンスのモニタリングとトラブルシューティング (186 ページ)

## 機能の概要と変更履歴

### 要約データ

該当製品または機能エリア	すべて
該当プラットフォーム	ASR 5500 VPC-SI VPC-DI
機能のデフォルト	無効：設定が必要
このリリースでの関連する変更点	N/A
関連資料	<ul style="list-style-type: none"><li>• <i>ASR 5500 System Administration Guide</i></li><li>• <i>Command Line Interface Reference</i></li><li>• VPC-DI システム管理ガイド</li><li>• VPC-SI System アドミニストレーションガイド</li></ul>

### マニュアルの変更履歴

改訂の詳細	リリース
オン/オフ機能ライセンスのスマート ライセンシング サポートが追加されました。	21.6

改訂の詳細	リリース
最初の導入。	21.3

## スマートソフトウェアライセンスング

### 概要

StarOS 21.3以降のリリースでは、スマートライセンスングがサポートされています。スマートライセンスングは、シスコのソフトウェア資産の購入、展開、管理をシンプル化するクラウドベースのライセンスングのアプローチです。権限付与はCisco Commerce Workspace (CCW) を介したシスコアカウントを通じて購入され、使用できるようにバーチャルアカウントにすぐに取り込まれます。これにより、あらゆるデバイスにライセンスファイルをインストールする必要がなくなります。スマートライセンスングが有効化されている製品では、使用状況のレポートがシスコに直接通知されます。シスコのソフトウェアライセンス (Cisco Smart Software Manager (CSSM)) を管理するために、お客様が1つの場所を使用できます。ライセンスの所有権と使用状況に関する情報をすぐに利用でき、使用状況やビジネスニーズに基づいて的確な購入判断ができます。

シスコスマートライセンスングの詳細については、<https://www.cisco.com/c/en/us/buy/smart-accounts/software-licensing.html> を参照してください。

### 従来のライセンスングとスマートライセンスングの比較

シスコでは、従来のライセンスングとスマートソフトウェアライセンスングの2種類のライセンスモデルを採用しています。**従来のライセンスング**は、製品アクティベーションキー (PAK) をシスコ製品にインストールすることによって、ソフトウェアアクティベーションで構成されます。製品アクティベーションキーは、他のシスコ機器と同様の方法で発注し、シスコ製品の機能セットのライセンスファイルを取得するために使用する購入可能品目です。**スマートソフトウェアライセンスング**は、ライセンスレポートを承認および提供するいくつかのツールを使用して、エンドツーエンドプラットフォームのクラウドベースのライセンスングです。StarOS に組み込まれたスマートソフトウェアライセンスング機能により、製品登録、承認が完了し、エンドカスタマーがレポートサービスを利用できるようになります。

### 評価期間

使用中のすべてのライセンスには、90日間の評価期間が付与されます。この期間中、機能ライセンスは制限なく使用でき、最大1つのカウントライセンスを使用できます。評価期間は、システムがCSSM または Cisco.com に正常に登録されると終了します。この90日の期間が満了すると、ライセンス付与された機能がブロックされます。

### リリース 21.6 のスマートライセンシング

21.6 以降のリリースでは、スマートライセンシングはオン/オフ機能ライセンスをサポートしています。それより前のリリースのスマートライセンシングでは、カウントライセンスのみがサポートされていました。

StarOS は、オン/オフ機能ライセンスでライセンスの強制を行います。各オン/オフ機能ライセンスは、それらのオン/オフ機能を使用する可能性があるサービスライセンスに関連付けられています。オン/オフライセンスのコンプライアンス違反（OOC）が検出された場合、次の条件に従って、対応するサービスの新しいコールがドロップされます。

- 各オン/オフ機能ライセンスには、90 日の猶予（評価）期間が与えられます。この間、システムは有効なライセンスが利用できないことを通知する SNMP トラップを生成します。OOC を解決するには、この機能のライセンスの購入と登録、または機能の無効化などの是正措置が必要です。
- 90 日の猶予期間後もこの機能が OOC である場合、StarOS は各ライセンスの事前定義されたポリシーに基づいて OOC 状態を強制します。強制が必要な場合は、オン/オフライセンスに対応するサービスの新しいコールがドロップされます。

使用中のスマートライセンスの強制に関する詳細を表示するために、次の 2 つの新しいコマンドが導入されました。

```
show license enforcement policy
show license enforcement status [ allowed | blocked ] [ feature | service ]
```

### リリース 21.3 のスマートライセンシング

リリース 21.3 でスマートライセンシングが有効になっている場合、従来のライセンシングとスマートライセンシングの両方が使用されている混合モードで動作します。

- 従来（PAK ベース）のライセンスは、引き続きオン/オフ機能に使用されます。
- スマートライセンシングは、各サービスをカウントするために使用されます。リリース 21.3 では、「権限付与タグ」に記載されているカウントライセンスのみがスマートライセンシングによって制御されます。他のすべてのカウントライセンスは、従来（PAK ベース）のライセンスを使用します。

## Cisco Smart Software Manager

Cisco Smart Software Manager（CSSM）を使用すると、ソフトウェアライセンスとスマートアカウントを単一のポータルから管理できます。このインターフェイスでは、製品の有効化、権限付与の管理、ソフトウェアの更新やアップグレードが可能です。登録プロセスを完了するには、機能しているスマートアカウントが必要です。Cisco Smart Software Manager にアクセスするには、こちら <https://software.cisco.com> をご覧ください。

## スマートアカウントおよびバーチャルアカウント

スマートアカウントでは、スマート対応のすべての製品と権限付与を1つの場所で管理します。これにより、シスコソフトウェアの調達、展開、およびメンテナンスを迅速に行うことができます。スマートアカウントを作成するには、要求元の組織を代表する権限が必要です。送信後、要求は簡単な承認プロセスを経由します。

バーチャルアカウントは、スマートアカウント内のサブアカウントとして存在します。バーチャルアカウントは、組織の配置、ビジネス機能、地域、または定義された階層に基づくお客様定義の構造体です。これらはスマートアカウント管理者によって作成および管理されます。

スマートアカウントの詳細、設定、または管理については、<https://software.cisco.com>を参照してください。

## Cisco スマートアカウントの要求

Cisco スマートアカウントは、スマートライセンスが有効になっているすべての製品が保管されているアカウントです。Cisco スマートアカウントを使用すると、デバイスのライセンスの管理とアクティブ化し、ライセンス使用状況のモニター、シスコライセンスの購入の追跡を行います。透過的なアクセスにより、スマートライセンス製品をリアルタイムで表示できます。IT 管理者は、Smart Software Manager を使用して、組織のスマートアカウント内のライセンスとアカウントユーザーを管理できます。

**ステップ 1** ブラウザのウィンドウに次の URL を入力します。

`https://software.cisco.com`

**ステップ 2** クレデンシアルを使用してログインし、[Administration] 領域で [Request a Smart Account] をクリックします。

[Smart Account Request] ウィンドウが表示されます。

**ステップ 3** [Create Account] で、次のいずれかのオプションを選択します。

- [Yes, I have authority to represent my company and want to create the Smart Account] : このオプションを選択した場合は、組織を代表して製品とサービスの資格、ユーザー、およびロールを作成し、管理する権限に同意したことになります。
- [No, the person specified below will create the account] : このオプションを選択した場合は、スマートアカウントを作成する担当者の電子メールアドレスを入力する必要があります。

**ステップ 4** [Account Information] で次の手順を実行します。

- a) [Account Domain Identifier] の横にある [Edit] をクリックします。
- b) [Edit Account Identifier] ダイアログボックスで、ドメインを入力し、[OK] をクリックします。デフォルトでは、ドメインはアカウントを作成する担当者の電子メールアドレスに基づいており、このアカウントを所有する企業に所属している必要があります。
- c) [Account Name] に入力します（通常は会社名）。

ステップ5 [Continue] をクリックします。

スマートアカウント要求は、アカウントドメイン識別子によって承認されるまで保留中の状態になります。承認後、設定プロセスを実行するための手順を含む電子メールの確認が送信されます。

## ソフトウェアタグと権限付与タグ

ライセンスを識別、レポート、および強制するために、次のソフトウェアおよび権限付与のタグが作成されています。

### ソフトウェアタグ

ソフトウェアタグは、デバイス上の各ライセンス可能なソフトウェア製品または製品スイートを一意に識別します。StarOS には、次のソフトウェアタグがあります。

製品タイプの説明	ソフトウェアタグ
ASR5500 ASR-5500 マルチメディアコアプラットフォーム	regid.2017-02.com.cisco.ASR5500,1.0_401f2e9e-67fd-4131-b61d-6e229d13a338
VPC_SI 仮想化パケットコア (単一インスタンス)	regid.2017-02.com.cisco.VPC_SI,1.0_dcb12293-10c0-4e90-b35e-b10a9f8bfac1
VPC_DI 仮想化パケットコア (分散型インスタンス)	regid.2017-02.com.cisco.VPC_DI,1.0_5cb68f91-c1d6-48d6-9482-e9750203f5e6

### 権限付与サービスタグ

次の権限付与タグは、各サービスのタイプで使用されているライセンスを識別します。

TagId	サービスタイプの説明	権限付与タグ
0	ASR5K-00-PW10GTWY PDN GW バンドル、10K セッション	regid.2017-02.com.cisco.ASR5K-00-PW10GTWY, 1.0_85a577a1-017d-4dc6-8bf7-4fac7c4d465a
1	ASR5K-00-SG10 SAE GW バンドル、10K セッション	regid.2017-02.com.cisco.ASR5K-00-SG10, 1.0_e87d1e76-e34c-4699-94ad-68eafa2da8b2
2	ASR5K-00-GN10SESS GGSN ソフトウェアライセンス、10K セッション	regid.2017-02.com.cisco.ASR5K-00-GN10SESS, 1.0_bf7eb1d7-35bf-40c6-8763-a4770f5ebf64

TagId	サービスタイプの説明	権限付与タグ
3	ASR5K-00-SW10LIC GW バンドル、10K セッションの提供	regid.2017-02.com.cisco. ASR5K-00-SW10LIC, 1.0_ca9d09ee-5e72-4695-97ef-d5117037cfb4
4	ASR5K-00-ME10LIC MME バンドル、10K セッション	regid.2017-02.com.cisco. ASR5K-00-ME10LIC, 1.0_37a452c9-ff23-41fc-b96c-c8b2eb09254b
5	ASR5K-00-EG10S-K9 EPDG、10k セッション	regid.2017-02.com.cisco. ASR5K-00-EG10S-K9, 1.0_5c625d0f-8b2c-4c44-970b-9668a072890f
6	ASR5K-00-SM10S-K9 SaMOG GW 10k セッション	regid.2017-02.com.cisco. ASR5K-00-SM10S-K9, 1.0_289f4807-fb6e-4750-b047-a9d57e7b045f

### 権限付与機能タグ

次の権限付与タグは、各機能のタイプで使用されているライセンスを識別します。

TagId	機能の説明	権限付与タグ
0	ASR5K-00-CS01GRET GRE インターフェイス トンネリング	regid.2017-11.com.cisco. ASR5K-00-CS01GRET, 1.0_284ba242-a2ef-4ad0-b445-b62ed3887d3b
1	ASR5K-00-CS01MPLS StarOS 上の MPLS	regid.2017-11.com.cisco. ASR5K-00-CS01MPLS, 1.0_1b1221cd-39de-4113-8709-dad022c0a063
2	ASR5K-00-GN01REC セッションリカバリ	regid.2017-11.com.cisco. ASR5K-00-GN01REC, 1.0_88542f00-077b-4ed0-9698-a3cec699a047
3	ASR5K-00-GN01ICSR シャーシ間セッションリカバリ	regid.2017-11.com.cisco. ASR5K-00-GN01ICSR, 1.0_d0bca3e3-1218-4e91-98f6-207a20a5401a
4	ASR5K-00-EP01VLE EPC Gw VoLTE の機能拡張	regid.2017-11.com.cisco. ASR5K-00-EP01VLE, 1.0_a6c3d615-0b37-4f4a-8c34-3090ed5e265c
5	ASR5K-00-EP01ICSE データとコントロールプレーンの ICSR 拡張リカバリ	regid.2017-11.com.cisco. ASR5K-00-EP01ICSE, 1.0_bf87e837-003e-4832-b29f-ce0416699b01
6	ASR5K-00-PWXICSRP ICSR/SR のパフォーマンスの向上	regid.2017-11.com.cisco. ASR5K-00-PWXICSRP, 1.0_079a8bd2-a97d-4d12-bc5a-b0f70eec6c26



TagId	機能の説明	権限付与タグ
7	ASR5K-00-CS01I-K9 StarOS 上の IPSec	regid.2017-11.com.cisco. ASR5K-00-CS01I-K9, 1.0_a6e5539c-ab69-4303-b2a8-e71c03ac2a08
8	ASR5K-00-EG01RS ePDG Reselection	regid.2017-11.com.cisco. ASR5K-00-EG01RS, 1.0_b7f0b7d1-0d68-4223-abb1-381c3ac968ec

## スマートライセンスの設定

作業を開始する前に、次を確認してください。

- <https://software.cisco.com> でスマートライセンス アカウントを作成した。
- スマートアカウント/バーチャルアカウントの一部として作成した製品インスタンス登録トークンを使用して製品を <https://software.cisco.com> で登録した。
- StarOS システムと CSSM サーバーまたは Cisco.com 間の通信パスを有効にした。

### スマートライセンスの有効化

デフォルトでは、StarOS ではスマートライセンスは無効になっています。スマートライセンスを有効にするには、次のコンフィギュレーションモード コマンドを入力します。

```
config
  license smart enable
end
```

次のコマンドを入力して設定を確認します。

```
show configuration | grep license
```

### シスコへのデバイスの登録

<https://software.cisco.com> で製品を登録したときに提供された ID 製品インスタンス登録トークンを使用し、次の Exec モードコマンドでシステムを登録します。

```
license smart register idtoken token
```

これで、システムは権限付与の使用数を CSSM サーバーに自動的に報告し、また、コンプライアンスステータスを受信するようになります。これにより、システムは「評価モード」からも削除されます。

コンプライアンスステータスを表示するには、次の Exec モードコマンドのいずれかを入力します。

```
show license status
show license summary
show license statistics
```

システムの登録が 180 日ごとに自動的に更新されます。必要に応じて、次の Exec モードコマンドを使用して、登録情報を手動で更新します。

```
license smart renew id
```

システムのライセンス承認は 30 日ごとに自動で更新されます。必要に応じて、次の Exec モードコマンドを使用して、ライセンス承認を手動で更新します。

```
license smart renew auth
```

デバイスの登録を解除するには、次の Exec モードコマンドを入力します。

```
license smart deregister
```

### コンプライアンス違反の処理

特定の SKU のバーチャルアカウントに十分なライセンスがない場合、CSSM はコンプライアンス違反 (OOC) メッセージをデバイスに送信します。システムは OOC 状態がクリアされるまで、追加のセッション許可を停止します。デバイスが承認済み応答を受信すると、OOC 状態がクリアされます。

## スマートライセンシングのモニタリングとトラブルシューティング

次の Exec モードコマンドを入力して、スマートライセンスの設定を確認します。

```
show configuration | grep license
```

次の Exec モードコマンドは、スマートライセンスに関する情報を表示します。

```
show license { all | enforcement | eval-period | smart-tags | statistics  
| status | summary | tech-support | udi | usage } [ | { grep grep_options  
| more } ]
```

注：

- **all** : ステータスの表示、使用状況の表示、UDI の表示、およびスマートライセンスエージェントのバージョンを含む情報のスパースセットを表示します。
- **enforcement { policy | status [ allowed | blocked ] [ feature | service ] }** : 適用された実施ポリシー、またはスマートライセンスの現在の適用ステータスを表示します。ステータス情報をフィルタリングして、現在許可またはブロックされているライセンスのみ、またはタイプ (機能ライセンスまたはサービスライセンス) ごとに表示できます。
- **eval-period** : 評価期間に関する情報を表示します。ライセンスには、登録されるまで 90 日の評価期間が付与されます。
- **smart-tags [ feature | service ]** : 現在サポートされている機能とサービス、および対応するスマート権限付与タグを表示します。
- **statistics** : 個々の機能ライセンスのステータスを表示します。

- **status** : スマートライセンスのステータス情報を表示します。
- **summary** : スマートライセンスのステータスの概要を表示します。
- **tech-support** : スマートライセンスの問題のデバッグに役立つ情報を表示します。
- **udi** : すべての一意のデバイス ID (UDI) の詳細を表示します。
- **usage** : 現在使用されているすべての権限付与の使用状況情報を表示します。

スマートライセンスの問題を追跡するために、次の SNMP アラーム (トラップ) が生成されます。

- 1349 : starSmartLicenseServiceOOC / starSmartLicenseServiceOOClear
- 1350 : starSmartLicenseEvalMode/starSmartLicenseEvalModeClear
- 1351 : starSmartLicenseEvalModeExpire/starSmartLicenseEvalModeExpireClear
- 1352 : starSmartLicenseCSSMConnnectionFail/starSmartLicenseCSSMConnectionFailClear



(注) これらの SNMP MIB アラームの詳細については、『*SNMP MIB Reference*』を参照してください。

## スマートライセンシングのバルク統計情報

スマートライセンシングのサポートでは、次の統計情報が提供されます。

- **current\_sl\_state** : スマートライセンスの現在の状態 (有効または無効)。
- **current\_sl\_mode** : スマートライセンスの現在のモード (評価または登録済み)。
- **total\_sl\_enabled** : スマートライセンスが有効化された合計回数。
- **total\_sl\_disabled** : スマートライセンスが無効化された合計回数。
- **mode** : 製品が評価および登録済みモードで登録されている場合の、各サービスタイプのカウンタ。
- **tagid** : 各サービスタイプの Tagid。
- **cur\_call\_policy** : 各サービスタイプの新しいセッションの現在のコールポリシー (ALLOW または BLOCK)
- **cur\_call\_count** : 特定のサービスタイプの製品全体についてカウントされた、現在のセッション数およびコール数。
- **max\_call\_count** : 特定のサービスタイプの製品全体についてカウントされたセッションおよびコールの最大数。

- `last_lic_count` : 特定のサービスタイプのシスコライセンス (CSSM) に最後に報告されたライセンス数。
- `max_lic_count` : この時点までに特定のサービスタイプのシスコライセンス (CSSM) に報告されたライセンスの最大数。
- `policy_change_block_calls` : 使用可能なライセンスが不足しているために、特定のサービスタイプが新しいセッションをブロックした回数。



## 第 13 章

# システムのモニタリング

この章では、コマンドラインインターフェイス（CLI）にある **show** コマンドを使用して、システムのステータスとパフォーマンスをモニタリングする方法について説明します。これらのコマンドには関連するキーワードが多数あります。それらのキーワードを使用すると、現在のソフトウェア設定からコールのアクティビティやステータスに至るまで、システムのすべての側面に関する有益な情報が得られます。

この章で説明するキーワードの選択は、システムのモニタリングに関する最も有益で詳細な情報の提供を目的としています。これらのキーワードやその他の **show** コマンドのキーワードの詳細については、『*Command Line Interface Reference*』の「*Exec Mode show Commands*」の章を参照してください。



(注) VPC-DI または VPC-SI の仮想マシン（VM）は、それが実行されているハイパーバイザや市販の（COTS）サーバーを認識しません。ハイパーバイザと COTS サーバーのステータスをモニターするには、このシステムのコンポーネントに付属しているユーザーマニュアルを参照してください。



**重要** リリース 21.1 以降では、グローバル コンフィギュレーション モードにある間に **do show** コマンドを使用してすべての Exec モードの **show** コマンドを実行します。**show** コマンドを実行するためにコンフィギュレーション モードを終了する必要はありません。パイプ文字 | は、コマンドが Exec モードで有効である場合にのみ使用できます。

- [SNMP 通知（190 ページ）](#)
- [システムのステータスとパフォーマンスのモニタリング（190 ページ）](#)
- [統計情報とカウンタのクリア（192 ページ）](#)

## SNMP 通知

CLIに加えて、システムはステータスとアラーム状態を示す Simple Network Management Protocol (SNMP) 通知をサポートしています。これらの通知の詳細なリストについては、『*SNMP MIB Reference*』を参照してください。

## システムのステータスとパフォーマンスのモニタリング

このセクションには、システム内のタスク、マネージャ、アプリケーション、およびその他のソフトウェアコンポーネントのステータスをモニターするために使用されるコマンドが含まれています。ほとんどのコマンドの出力に関する説明は、『*Statistics and Counters Reference*』に記載されています。

表 10: システムのステータスとパフォーマンスのモニタリングコマンド

手順は次のとおりです。	次のコマンドを入力します。
<b>管理情報を表示</b>	
現在の管理ユーザアクセスを表示	
現在システムにログオンしているすべての管理ユーザーのリストを表示	<b>show administrators</b>
管理ユーザーが作業しているコンテキスト、管理ユーザーが CLI にアクセスしている IP アドレス、およびシステムによって生成された ID 番号を表示	<b>show administrators session id</b>
システムに設定されているローカルユーザー管理アカウントに関連する情報を表示	<b>show local-user verbose</b>
ローカルユーザー管理アカウントの統計情報を表示	<b>show local-user statistics verbose</b>
CLI セッションに関連する情報を表示	<b>show cli</b>
<b>システム稼働時間を決定</b>	
システム稼働時間を表示 (前回の再起動からの経過時間)	<b>show system uptime</b>
<b>NTP サーバーのステータスを表示</b>	
NTP サーバーのステータスを表示	<b>show ntp status</b>
<b>システムリソースを表示</b>	
すべてのシステムリソースを表示 (CPU リソースや作成されたマネージャの数など)	<b>show resources [ cpu ]</b>
<b>システムアラームを表示</b>	
現在未処理のすべてのアラームに関する情報を表示	<b>show alarm outstanding all verbose</b>

手順は次のとおりです。	次のコマンドを入力します。
システムアラームの統計情報を表示	<b>show alarm statistics</b>
<b>輻輳制御の統計情報を表示</b>	
輻輳制御の統計情報を表示	<b>show congestion-control statistics</b>
<b>リモート管理の統計情報を表示</b>	
SNMP 通知の統計情報を表示	<b>show snmp notifies</b>
SNMP アクセスの統計情報を表示	<b>show snmp accesses</b>
SNMP トラップの履歴を表示	<b>show snmp trap history</b>
SNMP トラップの統計情報を表示	<b>show snmp trap statistics</b>
<b>ポートカウンタを表示</b>	
特定のポートのデータリンクカウンタを表示	<b>show port datalink counters slot#/port#</b>
特定のポートのポート ネットワーク プロセッサ ユニット (NPU) カウンタを表示	<b>show port npu counters slot#/port#</b>
<b>システム情報とネットワークインターフェイスを表示</b>	
システムコンポーネント、ストレージデバイス、およびネットワーク インターフェイスに関する情報を表示	<b>show hardware</b>
<b>カード情報と統計情報を表示</b>	
すべてのカードまたは特定のスロット/ポート (VPC の場合、スロットは VM) にあるカードの診断を表示	<b>show card diag slot/port</b>
すべてのカードまたは特定のスロット/ポート (VPC の場合、スロットは VM) にあるカードの詳細情報を表示	<b>show card info slot/port</b>
すべてのカードまたは VM の動作ステータスを表示	<b>show card table</b>
起動設定 (param.cfg) ファイルのコンテンツを表示 [VPC-DI]	<b>show cloud configuration</b>
設置されているハードウェアに関する情報と、システム内の特定のカードまたはすべてのカードに対して最適かどうかを表示 [VPC-DI]	<b>show cloud hardware</b>
特定のカードに関連する VPC-DI I ネットワークに関するモニター対象統計情報を表示 [VPC-DI]	<b>show cloud monitor di-network</b>



---

**重要** 使用可能なコマンドまたはキーワードと変数は、プラットフォームのタイプ、製品のバージョン、およびインストールされているライセンスによって異なります。

---



---

**重要** 一部のコマンドでは、プラットフォームのタイプによって出力が異なります。

---

## 統計情報とカウンタのクリア

新しい情報を収集するために、定期的に統計情報とカウンタをクリアする必要がある場合があります。システムは、グループ化（PPP、MIPHA、MIPFA など）に基づいて統計情報とカウンタをクリアする機能を提供します。

統計情報とカウンタは、CLI **clear** コマンドを使用してクリアできます。このコマンドの使用方法的詳細については、『*Command Line Interface Reference*』の「*Exec Mode Commands*」の章を参照してください。





## 第 14 章

# モニタープロセスのリスト

- [機能の概要と変更履歴 \(193 ページ\)](#)
- [機能説明 \(194 ページ\)](#)
- [モニターリングおよびトラブルシューティング \(194 ページ\)](#)

## 機能の概要と変更履歴

### 要約データ

該当製品または機能エリア	すべて
該当プラットフォーム	<ul style="list-style-type: none"><li>• ASR 5500</li><li>• VPC-DI</li><li>• VPC-SI</li></ul>
機能のデフォルト	有効、常時オン
このリリースでの関連する変更点	N/A
関連資料	<ul style="list-style-type: none"><li>• <i>ASR 5500 System Administration Guide</i></li><li>• <i>Command Line Interface Reference</i></li><li>• <i>Statistics and Counters Reference</i></li><li>• VPC-DI システム管理ガイド</li><li>• VPC-SI System アドミニストレーションガイド</li></ul>

### マニュアルの変更履歴



**重要** リリース 21.2 および N5.1 よりも前に導入された機能の改訂履歴の詳細は示していません。

改訂の詳細	リリース
最初の導入。	21.11

## 機能説明

モニタープロセスのリスト機能は、次の機能をサポートしています。

- 侵入を確認し、検出するための実行中プロセスの表示。
- ソフトウェアの改ざん防止を検出するためのソフトウェアの確認。
- セキュリティ上の決定事項の有効化。

新しく導入された CLI コマンドの **show process status** は、この機能をサポートしています。

## モニタリングおよびトラブルシューティング

この項では、機能のモニタリングとトラブルシューティングのサポートに使用できる CLI コマンドに関する情報を提供します。

### コマンドや出力の表示

この項では、この機能のサポートにおける **show** コマンドやその出力に関する情報を提供します。

#### **show process status**

この CLI コマンドの出力には、この機能のサポートに次のフィールドが含まれるようになりました。

- card - cpu
  - USER
  - PID
  - PPID
  - STARTED
  - %CPU
  - %MEM
  - COMMAND



---

(注) このコマンドは、セキュリティ管理者のみが実行できます。

---

`show process status`



## 第 15 章

# バルク統計情報

この章では、次の設定について説明します。

- [機能の概要と変更履歴 \(197 ページ\)](#)
- [収集サーバーとの通信の設定 \(198 ページ\)](#)
- [収集されたバルク統計情報データの表示 \(202 ページ\)](#)
- [SSD でのバルク統計情報サンプルの収集 \(202 ページ\)](#)
- [SFTP 公開キー認証 \(203 ページ\)](#)
- [バルク統計情報の手動収集と転送 \(204 ページ\)](#)
- [バルク統計のカウンタと情報のクリア \(204 ページ\)](#)
- [バルク統計情報スキーマの名称 \(204 ページ\)](#)
- [バルク統計情報のイベントログメッセージ \(208 ページ\)](#)

## 機能の概要と変更履歴

### 要約データ

該当製品または機能エリア	すべて
該当プラットフォーム	ASR 5500 VPC-DI VPC-SI
機能のデフォルト	無効：設定が必要
このリリースでの関連する変更点	N/A
関連資料	<ul style="list-style-type: none"><li>• <i>ASR 5500 System Administration Guide</i></li><li>• <i>Command Line Interface Reference</i></li><li>• <i>VPC-DI システム管理ガイド</i></li><li>• <i>VPC-SI System アドミニストレーションガイド</i></li></ul>

## マニュアルの変更履歴



(注) リリース 21.2 および N5.5 よりも前に導入された機能の改訂履歴の詳細は示していません。

改訂の詳細	リリース
SFTP 公開キーベースの認証のサポートが追加されました。詳細については、「 <i>SFTP</i> 公開キー認証のサポート」の項を参照してください。	21.24
バルク統計情報の設定パラメータを保持するオプションを使用して、設定済みのバルク統計情報のスキーマを置換または補足するための新しい機能が追加されました。  SSD 内のバルク統計情報のサンプルを収集するために、新しい機能が追加されました。詳細については、「 <i>SSD</i> でのバルク統計サンプルの収集」の項を参照してください。  この機能を有効にするため、 <b>bulkstat</b> グローバル コンフィギュレーションモード コマンドに <b>config [ schema   supplement ]</b> キーワードが追加されました。詳細については、「個別のバルク統計構成ファイルの設定」の項を参照してください。  <b>show configuration bulkstats brief</b> コマンドの出力が拡張され、スキーマを除くバルク統計情報設定の詳細が含まれるようになりました。	21.3
最初の導入。	21.2 よりも前

## 収集サーバーとの通信の設定

バルク統計情報の収集方法と管理方法を定義するには、2通りの設定方法を使用できます。「標準」設定では、システムがバルク統計情報ファイルに番号を自動的に割り当てることができます。オプションで、任意の設定方法で管理者が番号を指定できます。この章のコマンドの詳細と説明は、『*Command Line Interface Reference*』の、「*Bulk Statistics Configuration Mode Commands*」および「*Bulk Statistics File Configuration Mode Commands*」の章に記載されています。

### 標準の設定

この項の設定例では、バルク統計情報機能の基本的な動作を定義しています。次の設定例を使用して、統計情報収集サーバーと通信するようにシステムを設定します。

```
configure
  bulkstats mode
    schema name format format_string
    sample-interval time_interval
```

```

transfer-interval xmit_time_interval
limit mem_limit
exit
bulkstats collection
end

```

## オプションの設定

この項では、バルク統計情報コンフィギュレーションモードで使用できるオプションのコマンドについて説明します。具体的には、バルク統計情報をグループ化するためのバルク統計情報「ファイル」を設定できます。「ファイル」は、バルク統計情報スキーマ、配信オプション、および受信者の設定をグループ化するために使用されます。複数の「ファイル」を設定できるため、この機能により柔軟性が高まります。これは、さまざまな受信者に異なるスキーマを設定できるためです。

```

configure
bulkstats mode
  file number
    receiver ip_address { primary | secondary }
      [ mechanism { { { ftp | sftp } login user_name
        [ encrypted ] password pwd } | tftp } } ] }
    receiver mode { redundant | secondary-on-failure }
    remotefile format naming_convention [ both-receivers | primary-receiver
  | secondary-receiver ]
    header format header_format
    footer format footer_format
    exit
  schema_type schema format format_string
  sample-interval time_interval
  transfer-interval xmit_time_interval
  limit mem_limit
  exit
bulkstats collection
end

```



**重要** FTP はサポートされていません。SFTP は推奨される転送プロトコルです。

## バルク統計情報スキーマの設定

[標準の設定 \(198 ページ\)](#) および [オプションの設定 \(199 ページ\)](#) で説明されている各設定例で、スキーマのタイプと収集された統計情報を設定するために使用されるプライマリコマンドを次に示します。

```

configure
bulkstats mode
  schema_type schema format format_string

```

サポートされているスキーマ、使用可能な統計情報、および適切なコマンドシンタックスに関する詳細については、『*Command Line Interface Reference*』の「*Bulk Statistics Configuration Mode Commands*」および「*Bulk Statistics File Configuration Mode Commands*」の章を参照してください。

## 個別のバルク統計情報設定ファイルの設定

個別の接続先ファイルを設定してバルク統計情報サブモード設定を保存することができます。設定を確認するには、**show configuration bulkstats** コマンドを実行します。

バルク情報設定ファイルには、バルク統計情報サブモードでシステム設定ファイルに保存されていた設定が保存されます。

グローバル コンフィギュレーション モードで **bulkstats config** コマンドを実行すると、システム設定に個別のコンフィギュレーション ファイルが作成されます。

### configure

```
[no] bulkstats config [ schema | supplement ] url
end
```

オプションの **schema** キーワードを使用すると、指定されたファイルを使用してスキーマのみが置換されるため、サーバー設定を保持できます。オプションの **supplement** キーワードを使用すると、実行中のバルク統計情報の設定を、指定された設定ファイルの内容で補うことができます。これらのキーワードは、既存のファイルでのみ機能します。

*url* は、バルク統計情報設定ファイルの場所を指定します。接続先ファイルがすでに存在する場合は、新しいファイルに置き換えられます (**supplement** キーワードまたは **supplement** キーワードが使用されている場合を除く)。変更が完了した後に設定を保存した場合にのみ、新しいファイルが作成されます。Exec モードの **show configuration bulkstats** コマンドは、バルク統計情報設定モードの接続先ファイルが設定されている場合に、その URL を表示します。



**重要** **bulkstats config schema url** は手動設定よりも優先されます。スキーマに関しては、CLI を使用して設定を手動で追加、変更、または削除しても、変更は適用されません。

バルク統計設定ファイルを *url* からコピーして編集し、/flash にもう一度コピーできます。変更を適用するには、**bulkstats config** コマンドの **no** 形式を使用し、次に **bulkstats config** コマンドを再設定します。

**bulkstats config** コマンドが有効になっている場合、StarOS はシステム設定ファイルから既存のバルク統計情報サブモード設定を削除します。設定変更を保持するには、システム設定を保存する必要があります。

**bulkstats config** を使用して新しい接続先ファイルが有効になった後に無効にする場合、StarOS はファイルを削除しません。設定変更を保持するには、システム設定を保存する必要があります。





**重要** バルク統計情報設定の変更が完了したら、システム設定を保存して変更を保存する必要があります。**bulkstats config** コマンドが有効になっている場合は、バルク統計情報設定ファイルが更新されます。

## show bulkstats コマンドの使用

定義されたパラメータに関する情報を表示する Exec モードの **show bulkstats** コマンドはいくつかあります。

- **show bulkstats data** : 最大 4 つのファイルについて、統計情報の収集スキームに含まれている基準を表示します。収集されたバルク統計情報データの表示 (202 ページ) を参照してください。
- **show bulkstats schema** : 収集および送信統計情報を含む統計情報を収集するために使用されるスキームを表示します。設定の確認 (201 ページ) を参照してください。
- **show bulkstats variables** : スキーマ形式に組み込むことができるスキーマタイプ別に、使用可能な bulkstats 変数 (%variable%) を表示します。

また、**show configuration bulkstats brief** は、すべてのサーバー設定と、グローバルスコープでのバルク統計情報の設定を表示します。スキーマの設定は表示されません。

## 設定の確認

システム上でバルク統計情報のサポートを設定した後、保存する前に設定を確認できます。

この項の手順に従って、バルク統計情報の設定を確認します。これらの手順は、Exec モードのルートプロンプトを使用していることが前提になります。

次の Exec モードのコマンドを入力して、収集サーバーの通信とスキーマの設定を確認します。

### show bulkstats schemas

次に、コマンドの出力例を示します。

```
Bulk Statistics Server Configuration:
  Server State:                Enabled
  File Limit:                  6000 KB
  Sample Interval:            15 minutes (0D 0H 15M)
  Transfer Interval:          480 minutes (0D 0H 15M)
  Collection Mode:             Cumulative
  Receiver Mode:               Secondary-on-failure
  Local File Storage:         None
Bulk Statistics Server Statistics:
  Records awaiting transmission: 114
  Bytes awaiting transmission:   8092
  Total records collected:       59926
  Total bytes collected:         4190178
  Total records transmitted:    59812
  Total bytes transmitted:      4188512
  Total records discarded:      0
```

```

Total bytes discarded:          0
Last collection time required:  2 second(s)
Last transfer time required:    0 second(s)
Last successful transfer:       Wednesday December 7 12:14:30 EDT 2011
Last successful tx recs:        190
Last successful tx bytes:       13507
Last attempted transfer:        Wednesday December 7 12:14:30 EDT 2011

File 1
Remote File Format:             /users/ems/server/data/chicago/bulkstat%date%%time%.txt
File Header:                   "CHI_test %time%"
File Footer:                   ""
Bulkstats Receivers:
Primary: 192.168.0.100 using FTP with username administrator
Records awaiting transmission:  0
Bytes awaiting transmission:    0
Total records collected:        0
Total bytes collected:          0
Total records transmitted:      0
Total bytes transmitted:        0
Total records discarded:        0
Total bytes discarded:          0
Last transfer time required:    0 second(s)
No successful data transfers
No attempted data transfe

File 2 not configured

File 3 not configured

File 4 not configured

```

## コンフィギュレーションの保存

「設定の確認と保存」の章の説明に従って、設定を保存します。

## 収集されたバルク統計情報データの表示

システムは、収集されたが転送されていないデータを表示するためのメカニズムを備えています。このデータは「保留中のデータ」と呼ばれます。

次の Exec モードのコマンドを入力して、スキーマごとに保留中のバルク統計情報データを表示します。

```
show bulkstats data
```

上記のコマンドは、[オプションの設定 \(199ページ\)](#) で説明されているように設定した場合、リモートファイルの統計情報も表示します。

## SSD でのバルク統計情報サンプルの収集

show support details (SSD) コマンドの出力が収集され、トラブルシューティングを目的としてテクニカルアシスタンスセンター (TAC) に提供されます。SSD 内のバルク統計情報により、

お客様は分析に使用可能なバルク統計情報レコードをすぐに提供することができ、問題を迅速に解決することができます。

デフォルトでは、SSD にはバルク統計情報のサンプルは含まれていません。この機能は、グローバルコンフィギュレーションモードで **bulkstats ssd-samples** コマンドを使用して有効にできます。

```
config
  bulkstats ssd-samples { 1 | 2 }
end
```

以前に有効にしていた場合、**{ no | default } bulkstats ssd-samples** コマンドは SSD アーカイブ内のバルク統計情報のサンプルの収集を無効にします。バルク統計情報の各サンプルには、1つの転送履歴からのバルク統計情報が含まれています。現在、SSD アーカイブには最大2つのバルク統計情報のサンプルを含めることができます。サンプルファイルは、ファイル名 `ssd_bulkstats_file<bulkstat_file_number>_sample1.txt` (および `ssd_bulkstats_file<bulkstat_file_number>_sample2.txt`) の下にある `/var/tmp/bulkstats` の一時ストレージの場所に収集されます。

SSD アーカイブが一時ストレージ内に作成されている間に、バルク統計情報のサンプルで多量のストレージ容量が消費される可能性があります。その結果、SSD アーカイブの作成が失敗する場合があります。このようなシナリオでは、**show support details** コマンド内の **no-bulkstats** キーワードで SSD アーカイブからバルク統計情報サンプルを除外できます。

## SFTP 公開キー認証

### 機能説明

SFTP は、StarOS でのバルク統計転送用の公開キーベースの認証をサポートしています。より優れたセキュリティプラクティスを確実に順守するため、StarOS ベースの製品は外部サーバーにバルク統計を転送することを目的としてパスワードベースのメカニズムは使用しないでください。この機能により、パスワードの代わりに SSH キーを使用できます。バルク統計の転送メカニズムには、次の手順が含まれます。

1. RSA の秘密キーと公開キーのペアを生成します。

詳細については、『ASR 5500 システム管理ガイド』の「はじめに」の章で「SSH オプションの設定」>「外部サーバーへの SSH クライアントのログイン」>「SSH クライアントキーペアの生成」の項を参照してください。

2. 公開キーを外部のバルク統計サーバーにプッシュします。

詳細については、『ASR 5500 システム管理ガイド』の「はじめに」の章の「SSH オプションの設定」>「外部サーバーへの SSH クライアントのログイン」>「外部サーバーへの SSH クライアント公開キーのプッシュ」の項を参照してください。

手順 1 と 2 は既存のメカニズムであり、1 回だけ必要です。

3. 手順 1 と 2 で交換したキーを使用して、バルク統計情報ファイルを転送します。

詳細については、「*SFTP* 公開キー認証の設定」の項を参照してください。

## SFTP 公開キー認証

# バルク統計情報の手動収集と転送

スケジュール済みの間隔の範囲外でバルク統計情報を収集し、転送する必要がある場合があります。システムには、バルク統計情報の収集と転送を手動で開始できるコマンドが用意されています。

設定したサンプリング間隔の範囲外でバルク統計情報の収集を手動で開始するには、次の Exec モードのコマンドを入力します。

```
bulkstats force gather
```

設定した最大保存制限に到達する前にバルク統計情報の転送を手動で開始するには、次の Exec モードのコマンドを入力します。

```
bulkstats force transfer
```

# バルク統計のカウンタと情報のクリア

新しい情報を収集したり、すでに収集されているバルク統計情報を削除したりするために、バルク統計情報に関連するカウンタを定期的にクリアする必要がある場合があります。次の Exec モードのコマンドは、これらの機能のいずれかを実行するために使用できます。

```
clear bulkstats { counters | data }
```

**clear bulkstats data** コマンドは、転送されていない累積データをクリアします。これには、正常に転送されなかった「完了」ファイルが含まれます。

# バルク統計情報スキーマの名称

この項では、バルク統計情報の設定と表示に関連付けられている名称について説明します。

## 統計タイプ

次の統計タイプは『*Statistics and Counters Reference*』で定義されており、Exec モードの **show bulkstats variables** コマンドの出力に表示されます。

- **カウンタ** : カウンタでは、増分データが累積的に記録され、カウンタの制限に達するとロールオーバーされます。

- すべてのカウンタ統計情報は累積され、次のいずれかのメソッドでリセットされます。制限に達した際のロールオーバー、システムの再起動後、または `clear` コマンドの実行後。
- 制限はデータ型によって異なります。
- **ゲージ**：ゲージの統計情報は、単一の値を示します。定義されたタイムフレーム内における単一のポイントのスナップショット表現。値はある期間から次の期間へ繰り返される場合がありますが、ゲージはスナップショットごとに新しい値に変化します。制限はデータ型によって異なります。
- **情報**：このタイプの統計情報は、多くの場合、統計情報のセットを区別することを目的とした情報を提供します（たとえば、VPN 名または IP アドレス）。提供される情報のタイプは、データ型によって異なります。

次の統計タイプは、『*Statistics and Counters Reference*』のスプレッドシートに含まれており、元のユーザードキュメントを置き換えます。

- **増分**：増分のデータ型では、増分データが累積的に記録され、カウンタの制限に達するとロールオーバーされます。
  - すべての増分統計情報は累積され、次のいずれかのメソッドでリセットされます。制限に達した際のロールオーバー、システムの再起動後、または `clear` コマンドの実行後。
  - 制限はデータ型によって異なります。
- **ゲージ**：ゲージの統計情報は、単一の値を示します。定義されたタイムフレーム内における単一のポイントのスナップショット表現。値はある期間から次の期間へ繰り返される場合がありますが、ゲージはスナップショットごとに新しい値に変化します。制限はデータ型によって異なります。
- **プライマリキー**：このタイプの統計情報は、多くの場合、統計情報のセットを区別することを目的とした情報を提供します（たとえば、VPN 名または IP アドレス）。提供される情報のタイプは、データ型によって異なります。

## データ型

データ型は、統計情報によって提供される値のデータの形式を定義します。次のデータ型は、『*Statistics and Counters Reference*』、および Exec モードの `show bulkstats variables` コマンドの出力に表示されます。

- **Int32**：32 ビットの整数。ゼロ制限へのロールオーバーは 4,294,967,295 です。
- **Int64**：64 ビットの整数。ゼロ制限へのロールオーバーは 18,446,744,073,709,551,615 です。
- **Float**：小数点を含む数値。たとえば、1.345。
- **文字列**：1つのグループに含まれる一連の ASCII 英数字（通常は事前設定されています）。

## 主な変数

すべてのスキーマには、通常「キー変数」と呼ばれるいくつかの変数があります。これらのキー変数は、統計がどのオブジェクトに適用されるかを識別するためのインデックスマーカーを提供します。たとえば、カードスキーマのカード番号（変数 %card%）カードを一意に識別します。HA サービスの場合、キーは「%vpnname%」と「%servname%」です。この組み合わせで、HA サービスを一意に識別します。したがって、特定の測定間隔では、一意のキーごとに 1 行の統計情報が生成されます。

また、時間、日付、場所などを識別する複数の共通変数がスキーマ間で共有されています。これらの一般的な変数は、次の表で識別されます。

表 11: スキーマ全体の共通変数

変数	説明	統計タイプ	データタイプ
version-no	バルク統計情報ファイルのヘッダーで使用できる完全なバージョン情報が含まれています。	情報	文字列
uptime	ファイルを作成したシステムの合計稼働時間（秒単位）。	情報	Int32
host	ファイルを作成したシステムホスト名。	情報	文字列
ipaddr	dPv4 ドット付き 10 進表記のデフォルトの管理（ローカルコンテキスト）IP アドレス。使用可能なアドレスがない場合は、空の文字列が挿入されます。	情報	文字列
date	収集ファイルが作成された、YYYYMMDD 形式の UTC 日付。YYYY は年を表し、MM は月を表し、DD は日を表します。	情報	文字列
date3	収集ファイルが作成された、YYMMDD 形式の UTC 日付。YY は年を表し、MM は月を表し、DD は日を表します。	情報	文字列
time	収集ファイルが作成された、HHMMSS 形式の UTC 時間。HH は時間を表し、MM は分を表し、SS は秒を表します。	情報	文字列
time2	収集ファイルが作成された、HH:MM:SS 形式の UTC 時間。HH は時間を表し、MM は分を表し、SS は秒を表します。	情報	文字列

変数	説明	統計タイプ	データタイプ
time3	収集ファイルが作成された、HH:MM形式のUTC時間。HHは時間を表し、MMは分を表します。	情報	文字列
epochtime	1970年1月1日00:00:00 GMT以降の秒数。	情報	In32
schemas	このプラットフォームで使用可能なすべてのバルク統計情報スキーマを一覧表示します。	情報	文字列
schemas-delta	最後に出力されたスキーマリストを変更したすべてのバルク統計情報スキーマを一覧表示します。	情報	文字列
localdate	収集ファイルが作成された、YYYYMMDD形式の日付（ローカルタイムゾーンに調整済み）。YYYYは年を表し、MMは月を表し、DDは日を表します。	情報	文字列
localdate3	収集ファイルが作成された、YYYYMMDD形式の日付。YYYYは年を表し、MMは月を表し、DDは日を表します。日付はUTCではなく現地時間で表示されます。	情報	文字列
localtime	収集ファイルが作成された、HHMMSS形式の時間（ローカルタイムゾーンに調整済み）。HHは時間を表し、MMは分を表し、SSは秒を表します。	情報	文字列
localtime2	収集ファイルが作成された、HH:MM:SS形式の時間（ローカルタイムゾーンに調整済み）。HHは時間を表し、MMは分を表し、SSは秒を表します。	情報	文字列
localtime3	収集ファイルが作成された、HH:MM:SS形式の時間。HHは時間を表し、MMは分を表し、SSは秒を表します。時間はUTCではなく現地時間で表示されます。	情報	文字列
localtz	このプラットフォームに設定されているローカルタイムゾーン。	情報	文字列
localzoffset	ローカルタイムゾーンのUTC/GMTからのオフセット。Format = "+" または "-" HHMM。	情報	文字列
swbuild	StarOS バージョンのビルド番号。	情報	文字列

## バルク統計情報のイベントログメッセージ

stat ログイングファシリティは、特定の場所に設定されたバルク統計情報データの作成または書き込みのいずれかで、発生する可能性のあるエラーを診断するために役立ついくつかのイベントをキャプチャします。

次の表に、これらのイベントに関連する情報を示します。

表 12: バルク統計情報に関連するイベントのログイング

イベント	イベント ID	シビラティ (重大度)	その他の情報
ローカルファイルのオープンエラー	31002	警告	「Unable to open local file <i>filename</i> for storing data」
受信者のオープンエラー	31018	警告	「Unable to open url <i>filename</i> for storing bulk data」
受信者の書き込みエラー	31019	警告	「Unable to write to url <i>filename</i> while storing data」
受信側のクローズエラー	31020	警告	「Unable to close url <i>filename</i> while storing data」





## 第 16 章

# システム ログ

この章では、さまざまなタイプのロギングに関連するパラメータを設定する方法と、その内容を表示する方法について説明します。内容は次のとおりです。

- [機能の概要と変更履歴 \(209 ページ\)](#)
- [システムログのタイプ \(211 ページ\)](#)
- [イベントロギングパラメータの設定 \(212 ページ\)](#)
- [アクティブログの設定 \(217 ページ\)](#)
- [ファシリティの指定 \(218 ページ\)](#)
- [トレースロギングの設定 \(228 ページ\)](#)
- [モニターログの設定 \(229 ページ\)](#)
- [ロギング設定と統計情報の表示 \(230 ページ\)](#)
- [CLIを使用したイベントログの表示 \(230 ページ\)](#)
- [クラッシュログの設定と表示 \(231 ページ\)](#)
- [過剰なイベントロギングの削減 \(235 ページ\)](#)
- [ログのチェックポイントニング \(236 ページ\)](#)
- [ログファイルの保存 \(237 ページ\)](#)
- [イベント ID の概要 \(237 ページ\)](#)

## 機能の概要と変更履歴

### 要約データ

該当製品または機能エリア	すべて
該当プラットフォーム	<ul style="list-style-type: none"><li>• ASR 5500</li><li>• UGP</li><li>• VPC-DI</li><li>• VPC-SI</li></ul>

機能のデフォルト	有効
このリリースでの関連する変更点：	N/A
関連資料	<ul style="list-style-type: none"> <li>• <i>ASR 5500 System Administration Guide</i></li> <li>• <i>Command Line Interface Reference</i></li> <li>• VPC-DI システム管理ガイド</li> <li>• VPC-SI System アドミニストレーション ガイド</li> </ul>

### マニュアルの変更履歴



(注) リリース 21.2 および N5.5 よりも前に導入された機能の改訂履歴の詳細は示していません。

改訂の詳細	リリース
<p>この標準で定義されている RFC5424 と syslog メッセージングの標準をサポートするため、StarOS 内の syslog クライアントがこのリリースで更新されました。StarOS は、以前の RFC3164 メッセージフォーマットを引き続きサポートします。また、このリリースでは、複数のポートを使用して複数の syslog サーバーの IP アドレスを設定することもできます。</p> <p>(注) リリース 21.6 では、UDP のみを使用したトランスポート層のメッセージングがサポートされています。このリリースでは、TLS と TCP はサポートされていません。</p>	21.6
<p>管理者がイベント ID またはイベント ID の範囲に対してロギングを完全に無効にした場合、またはロギングのレベルをデフォルトのロギングレベル未満（エラーレベル）に変更した場合に通知を行うため、2つの新しいクリティカル CLI イベントログと 2つの新しい SNMP トラップが追加されています。このリリースでは、これらのイベントログとトラップはデフォルトで有効になっており、無効にすることはできません。詳細については、<a href="#">グローバルコンフィギュレーションモードのフィルタリング (215 ページ)</a> を参照してください。</p> <p>この機能の結果として、追加または変更されたコマンドはありません。</p> <p><b>show snmp trap statistics</b> コマンド出力が拡張され、ロギングイベントが無効になっているか、またはロギングレベルがデフォルトの（エラー）ロギングレベルよりも下に変更された場合に、イベント内に詳細が表示されるようになりました。</p>	21.3
最初の導入。	21.2 よりも前

# システムログのタイプ

システムで設定および表示できるログには、次の5つのタイプがあります。



**重要** すべてのイベントログをすべての製品で設定できるわけではありません。設定可能性は、使用されているハードウェアプラットフォームとライセンスによって異なります。

- **イベント**：イベントロギングを使用して、システムのステータスを判断し、システムで使用されているプロトコルとタスクに関する重要な情報をキャプチャできます。これは、すべてのコンテキスト、セッション、およびプロセスに適用されるグローバル機能です。
- **アクティブ**：アクティブログは、CLI instance-by-CLI インスタンスベースでオペレータが設定可能です。1つのCLI インスタンスで管理ユーザーによって設定されたアクティブログは、別のCLI インスタンスの管理ユーザーによって表示することはできません。各アクティブログは、システムに対してグローバルに設定されているものとは独立したフィルタプロパティと表示プロパティを使用して設定できます。イベントが生成されると、アクティブログがリアルタイムで表示されます。
- **トレース**：トレースロギングを使用して、接続されている特定のサブスクライバセッションで発生する可能性のある問題を迅速に分離できます。トレースは、特定のコール ID (callid) 番号、IP アドレス、モバイルステーション ID (MSID) 番号、またはユーザー名に対して実行できます。
- **モニター**：モニターロギングは、特定のセッションに関連付けられているすべてのアクティビティを記録します。この機能は、特定のサブスクライバのモニタリング機能に関する司法当局の要請を遵守するために使用できます。モニターは、サブスクライバの MSID またはユーザー名に基づいて実行できます。
- **クラッシュ**：クラッシュロギングには、システムソフトウェアのクラッシュに関する有用な情報が保存されています。この情報は、クラッシュの原因を特定するのに役立ちます。



**重要** ステートフルファイアウォールと NAT は、ロギングがファイアウォールに対して有効になっている場合に、画面上のさまざまなメッセージのロギングをサポートします。これらのログは、critical、error、warning、および debug などのさまざまなレベルで詳細なメッセージを提供します。ステートフルファイアウォールと NAT 攻撃ログは、攻撃によってドロップされたパケットの送信元 IP アドレス、宛先 IP アドレス、プロトコル、または攻撃タイプに関する情報も提供します。また、システムで設定されている場合は、syslog サーバーにも送信されます。ステートフルファイアウォールおよび NAT のロギングサポートの詳細については、『*PSF Administration guide*』または『*NAT Administration Guide*』の「Logging Support」の章を参照してください。

## イベントロギングパラメータの設定

システムは、ユーザー定義のフィルタに基づいてログを生成するように設定できます。フィルタは、システムがモニターするファシリティ（システムタスクまたはプロトコル）と、イベントエントリの生成をトリガーするシビラティ（重大度）レベルを指定します。

イベントログはシステムメモリに保存され、CLIを使用して表示できます。イベントロギング情報を保存する2つのメモリバッファがあります。最初のバッファには、アクティブなログ情報が保存されます。2番目のバッファには、非アクティブなロギング情報が保存されます。非アクティブバッファは一時的なリポジトリとして使用され、データを上書きせずにログを表示できます。ログは、手動による介入によってのみ非アクティブバッファにコピーされます。

各バッファには最大50,000のイベントを保存できます。これらのバッファがキャパシティに達すると、最も古い情報が削除され、最新の情報を保存するための領域が確保されます。

ログデータの損失を防ぐために、ネットワーク インターフェイスを介して `syslog` サーバーにログを送信するようにシステムを設定できます。




---

**重要** TACACS+ アカウンティング (CLI イベントロギング) は、合法的傍受ユーザー (priv レベル 15 および 13) に対して生成されません。

---

## イベントログフィルタの設定

Exec モードおよびグローバル コンフィギュレーション モード レベルでイベントログの内容をフィルタリングできます。詳細については、『*Command Line Interface Reference*』を参照してください。

### Exec モードのフィルタリング

これらのコマンドを使用すると、グローバル ロギング パラメータを変更することなく、ログに含まれるデータの量を制限できます。

Exec モードのコマンドを介してログをフィルタ処理するには、次の例に従います。

#### アクティブなフィルタリング

```
logging active [ copy runtime filters ] [ event-verbosity event_level ]
[ pdu-data format ] [ pdu-verbosity pdu_level ]
```

注：

- **copy runtime filters** : ランタイムフィルタをコピーし、そのコピーを使用して現在のロギングセッションをフィルタ処理します。
- **event-verbosity event\_level** : イベントのロギングで使用する *verbosity* のレベルを次のいずれかに指定します。

- *min* : イベントに関する最小限の情報を表示します。情報には、イベント名、ファシリティ、イベント ID、シビラティ (重大度) レベル、日付、および時刻が含まれます。
- *concise* : イベントに関する詳細情報を表示しますが、システム内にイベントの送信元を提供しません。
- *full* : イベントが生成されたシステム内の場所を識別する送信元情報など、イベントに関する詳細情報を表示します。
- **pdu-data format** : ログに記録された場合のデータユニットの出力形式を次のいずれかに指定します。
  - *none* : RAW 形式 (未フォーマット)
  - *hex* : 16 進数形式
  - *hex-ascii* : メインフレームのダンプと同様の 16 進数および ASCII
- **pdu-verbosity pdu\_level** : データユニットのログに使用する *verbosity* のレベルを 1 ~ 5 の整数で指定します。5 が最も詳細なものです。  
「設定の確認と保存」の章の説明に従って、設定を保存します。

#### インスタンスによるフィルタリングの無効化または有効化

```
logging filter active facility facility_level severity_level [ critical-info | no-critical-info ]
```

```
logging filter { disable | enable } facility facility { all | instance instance_number }
```

注 :

- **active** : アクティブプロセスのみにログオプションを設定するように指示します。
- **disable** : 特定のインスタンスまたはすべてのインスタンスのログを無効にします。このキーワードは、*aaamgr*、*hamgr*、および *sessmgr* ファシリティでのみサポートされています。
- **enable** : 特定のインスタンスまたはすべてのインスタンスのログを有効にします。このキーワードは、*aaamgr*、*hamgr*、および *sessmgr* ファシリティでのみサポートされています。デフォルトでは、*aaamgr*、*hamgr*、および *sessmgr* のすべてのインスタンスに対してログが有効になっています。
- **facility facility and level severity\_level** : どのシステムファシリティをどのレベルでログに記録するかを決定するログフィルタを設定します。詳細については、[ファシリティの指定 \(218 ページ\)](#) と [イベントのシビラティ \(重大度\) \(246 ページ\)](#) を参照してください。
- **all | instance instance\_number** : すべてのインスタンス、または *aaamgr*、*hamgr*、または *sessmgr* の特定のインスタンスに対して、ログを無効にするか有効にするかを指定し

ます。**show session subsystem facility facility** コマンドを実行して、特定のインスタンス番号を識別します。



---

(注) これらのキーワードは、**disable** および **enable** キーワードでのみサポートされています。

---

- **level severity\_level** : 次のリストからログに記録される情報のレベルを指定します。このレベルは、最高から最低の順に並べられます。
  - **critical** : エラーイベントを表示
  - **error** : エラーイベントおよびシビラティ (重大度) レベルが高いすべてのイベントを表示
  - **warning** : 警告イベントおよびシビラティ (重大度) レベルが高いすべてのイベントを表示
  - **unusual** : 異常イベントおよびシビラティ (重大度) レベルが高いすべてのイベントを表示
  - **info** : 情報イベントおよびシビラティ (重大度) レベルが高いすべてのイベントを表示
  - **trace** : トレースイベントおよびシビラティ (重大度) レベルが高いすべてのイベントを表示
  - **debug** : すべてのイベントを表示



---

(注) このキーワードは、**active** キーワードとの組み合わせでのみサポートされています。

---

- **critical-info** : 重要な情報のカテゴリ属性を持つイベントを表示するように指定します。これらのタイプのイベントの例は、システムプロセスとタスクが開始される際のブートアップ時に表示されます。これがデフォルトの設定です。  
**no-critical-info** : 重要な情報のカテゴリ属性を持つイベントを表示しないように指定します。



---

(注) これらのキーワードは、**active** キーワードとの組み合わせでのみサポートされています。

---



**重要** ファシリティの単一のインスタンスにおけるロギングを有効にするには、まずファシリティのすべてのインスタンスを無効にしてから、 (**logging filter disable facility facility all**) 次に特定のインスタンスのロギングを有効にする必要があります (**logging filter enable facility facility instance instance\_number**)。デフォルトの動作に復元するには、すべてのインスタンスのロギングを再度有効にする必要があります (**logging filter enable facility facility all**)。

Exec モードの **show instance-logging** コマンドを使用して、ファシリティごとに有効にされたインスタンスのインスタンス番号を表示できます。

## グローバル コンフィギュレーション モードのフィルタリング

Exec モードおよびグローバル コンフィギュレーション モード レベルでイベントログの内容をフィルタリングできます。

システムの実行時間イベントロギングパラメータを設定するには、次の例に従います。

### configure

```
logging filter runtime facility facility level report_level
logging display { event-verbosity | pdu-data | pdu-verbosity }
end
```

注：

- **facility facility and level severity\_level**：どのシステムファシリティをどのレベルでログに記録するかを決定するロギングフィルタを設定します。詳細については、[ファシリティの指定 \(218 ページ\)](#) と [イベントのシビラティ \(重大度\) \(246 ページ\)](#) を参照してください。
- ログに記録するすべてのファシリティに対して手順を繰り返します。

- オプション：**logging disable eventid** コマンドを追加して、イベント ID の制限を設定します。システムは、特定のイベント ID またはある範囲のイベント ID の送信を制限して、ログに記録されるデータの量が最も有用なレベルとなるよう最小限に抑える機能を提供します。追加のイベント ID またはイベント ID の範囲のロギングを無効にするには、この手順を繰り返します。

- 管理者が上記のコマンド (**logging disable eventid**) を使用してイベント ID またはイベント ID 範囲のイベントロギングを制限すると、システムは重大なイベントログ「cli 30999 critical」と、無効にされた特定のイベント ID またはイベント ID 範囲を持つ SNMP トラップ「1361 (DisabledEventIDs)」を生成します。

このリリースでは、これらのイベントログとトラップはデフォルトで有効になっており、無効にすることはできません。

- 管理者がロギングレベルを下げる (**logging filter runtime facility facility level report\_level** コマンドを使用して、デフォルトレベルである「error」より下にする) と、システムは重大なイベントログ「cli 30998 Critical」と、無効にされた特定のイベント ID またはイベント ID 範囲を持つ SNMP トラップ「1362 (LogLevelChanged)」を生成します。

このリリースでは、これらのイベントログとトラップはデフォルトで有効になっており、無効にすることはできません。

次の例は、イベントロギングまたはログレベルが変更された場合に生成されるトラップの CLI 出力を示しています。

```
[local]host# show snmp trap statistics
SNMP Notification Statistics:
...
Trap Name                               #Gen #Disc  Disable Last Generated
-----
...
DisabledEventIDs                        1     0     0  2017:05:11:15:35:25
LogLevelChanged                          2     0     0  2017:05:11:15:28:03

[local]host# show snmp trap history
There are x historical trap records (5000 maximum)

Timestamp                               Trap Information
-----
...
Thu May 11 15:28:03 2017 Internal trap notification 1362 (LogLevelChanged) Logging level
of facility resmgr is changed to critical by user #initial-config# context local privilege
level Security Administrator ttyname /dev/pts/0 address type IPV4 remote ip address
0.0.0.0
...
Thu May 11 15:35:25 2017 Internal trap notification 1361 (DisabledEventIDs) Event IDs
from 100 to 1000 have been disabled by user adminuser context privilege level
security administrator ttyname tty address type IPV4 remote ip address 1.2.3.4
...
Mon May 15 10:14:56 2017 Internal trap notification 1362 (LogLevelChanged) Logging level
of facility sitmain is changed to critical by user staradmin context local privilege
level Security Administrator ttyname /dev/pts/1 address type IPV4 remote ip address
161.44.190.27
```

「設定の確認と保存」の章の説明に従って、設定を保存します。

## syslog サーバの設定

### syslog アーキテクチャ

システムロギング (syslog) は、StarOS から UDP トランスポート層を介してイベント情報を生成し、一元化されたイベントメッセージコレクタに送信するアーキテクチャです。syslog はクライアント/サーバーアーキテクチャを使用します。

- **syslog クライアント** : StarOS 製品で実行されている一連のプロセスであり、イベントメッセージの送信デバイスとして動作します。
- **syslog サーバー** : StarOS 製品から送信されたイベントメッセージを受信するように設定された外部サーバーです。

StarOS 製品は、受信の確認応答を必要とせずに、syslog プロトコルを使用してイベントメッセージを転送します。システムは、syslog サーバーがメッセージを受信できるかどうかに関係なく、イベントメッセージを転送します。



## 外部 syslog サーバーへイベントメッセージを送信するようにするシステムの設定

実行時イベントロギングのフィルタで生成された情報は、永続ストレージ用の syslog サーバーに送信できます。



**重要** syslog サーバーに送信されるデータは、情報提供のために使用することを目的としています。課金やパフォーマンスのモニタリングなどの機能は、syslog には基づかないようにする必要があります。



**重要** システムはコンテキストごとに syslog サーバーを設定する柔軟性を備えていますが、ネットワークトラフィックからのログトラフィックを分離するために、すべてのサーバーをローカルコンテキストで設定することを推奨します。

Syslog サーバーを設定するには、次の例を使用します。

```
configure
context local
  logging syslog ip_address
end
```

注：

- `ip_address` は、ネットワーク上のシステムログサーバーの IP アドレスを、IPv4 のドット付き 10 進表記か、または IPv6 のコロンで区切られた 16 進表記で指定します。
- `logging syslog` コマンドでは、いくつかのオプションのキーワードを使用できます。詳細については、『*Command Line Interface Reference*』の「*Context Configuration Mode Commands*」の章を参照してください。
- 追加の syslog サーバーを設定する場合は、必要に応じてこの手順を繰り返します。設定可能な syslog サーバーの数に制限はありません。

詳細については、『*Command Line Interface Reference*』の `logging` コマンドを参照してください。

「設定の確認と保存」の章の説明に従って、設定を保存します。

## アクティブログの設定

アクティブログは、CLI `instance-by-CLI` インスタンスでオペレータが設定可能なイベントログです。1 つの CLI インスタンスで管理ユーザーによって設定されたアクティブログは、別の CLI インスタンスの管理ユーザーには表示されません。各アクティブログは、システムに対してグローバルに設定されているものとは独立したフィルタプロパティと表示プロパティを使用して設定できます。アクティブログは、生成時にリアルタイムで表示されます。

アクティブなログは、デフォルトではアクティブなメモリバッファに書き込まれません。アクティブなログをアクティブなメモリバッファに書き込むには、グローバル コンフィギュレーション モードで次のコマンドを実行します。

```
[local]host_name(config)# logging runtime buffer store all-events
```

アクティブなログがアクティブなメモリバッファに書き込まれると、すべての CLI インスタンスのすべてのユーザーが使用できるようになります。

次の例を使用して、グローバル コンフィギュレーション モードでアクティブロギングを設定します。

```
[local]host_name(config)# logging filter runtime facility facility level
report_level
```

注：

- どのシステムファシリティをどのレベルでログに記録するかを決定するロギングフィルタを設定します。詳細については、[ファシリティの指定 \(218 ページ\)](#) と [イベントのシビラティ \(重大度\) \(246 ページ\)](#) を参照してください。
- ログに記録するすべてのファシリティに対して手順を繰り返します。
- オプション：**logging disable eventid** コマンドを追加して、イベント ID の制限を設定します。システムは、特定のイベント ID またはある範囲のイベント ID の送信を制限して、ログに記録されるデータの量が最も有用なレベルとなるよう最小限に抑える機能を提供します。追加のイベント ID またはイベント ID の範囲のロギングを無効にするには、この手順を繰り返します。
- Exec モードの **logging active** コマンドでは、いくつかのキーワードオプションと変数を使用できます。詳細については、『*Command Line Interface Reference*』の「*Exec Mode Commands*」の章を参照してください。

必要なすべての情報が収集されたら、Exec モードで次のコマンドを入力して、アクティブなログの表示を停止できます。

```
no logging active
```

## ファシリティの指定



**重要** ロギングに使用できる実際のファシリティは、プラットフォームタイプ、StarOS のバージョン、およびインストールされている製品ライセンスによって異なります。

次のファシリティは、イベントデータをロギングするように設定できます。

- **a10** : A10 インターフェイス ファシリティ
- **a11** : A11 インターフェイス ファシリティ
- **a11mgr** : A11 マネージャファシリティ
- **aaa-client** : 認証、許可、およびアカウンティング (AAA) クライアントファシリティ

- **aaamgr** : AAA マネージャのロギングファシリティ
- **aaaproxy** : AAA プロキシファシリティ
- **aal2** : ATM アダプテーション レイヤ 2 (AAL2) プロトコルのロギング ファシリティ
- **acl-log** : アクセスコントロールリスト (ACL) のロギングファシリティ
- **acsctrl** : アクティブ チャージング サービス (ACS) コントローラファシリティ
- **acsmgr** : ACS マネージャファシリティ
- **afctrl** : ファブリックコントローラファシリティ [ASR 5500 のみ]
- **afmgr** : Fabric Manager のロギングファシリティ [ASR 5500 のみ]
- **alarmctrl** : アラームコントローラファシリティ
- **alcap** : Access Link Control Application Part (ALCAP) プロトコルのロギングファシリティ
- **alcapmgr** : ALCAP マネージャのロギングファシリティ
- **all** : すべてのファシリティ
- **bfd** : Bidirectional Forwarding Detection (BFD) プロトコル
- **bgp** : ボーダー ゲートウェイ プロトコル (BGP) ファシリティ
- **bindmux** : IPCF BindMux-Demux マネージャのロギングファシリティ
- **bngmgr** : ブロードバンドネットワーク ゲートウェイ (BNG) Demux マネージャのロギングファシリティ
- **bssap+** : SGSN と MSC/VLR (2.5G および 3G) 間のログインインターフェイス用のベースステーションサブシステムアプリケーション部品+プロトコルファシリティ
- **bssgp** : ベースステーションサブシステムの GPRS プロトコルログインファシリティは、SGSN と BSS 間の情報の交換を処理します (2.5G のみ)
- **callhome** : Call Home アプリケーションのロギングファシリティ
- **cap** : プリペイドアプリケーション (2.5G および 3G) で使用されるプロトコル用 CAMEL Application Part (CAP) のロギングファシリティ
- **cbsmgr** : セルブロードキャストサービス (CBS) のロギングファシリティ [HNBGW]
- **cdf** : チャージング データ ファンクション (CDF) のロギングファシリティ
- **cfctrl** : コンテンツ フィルタリング コントローラのロギングファシリティ
- **cfmgr** : コンテンツ フィルタリング マネージャのロギングファシリティ
- **cgw** : 統合アクセスゲートウェイ (CGW) のロギングファシリティ
- **cli** : コマンドラインインターフェイス (CLI) のロギングファシリティ
- **cmp** : 証明書管理プロトコル (IPSec) のロギングファシリティ

- **confdmgr** : ConfD Manager Proctlet (netconf) のロギングファシリティ
- **connectedapps** : SecGW ASR 9000 oneP 通信プロトコル
- **connproxy** : コントローラプロキシのロギングファシリティ
- **credit-control** : Credit Control (CC) ファシリティ
- **csp** : カード/スロット/ポート コントローラ ファシリティ
- **css** : コンテンツ サービス セレクション (css) ファシリティ
- **css-sig** : CSS RADIUS シグナリングファシリティ
- **cx-diameter** : Cx Diameter メッセージファシリティ [CSCF <--> HSS]
- **data-mgr** : データ マネージャ フレームワーク のロギングファシリティ
- **dcardctrl**: IPSec のドーターカードコントローラのロギングファシリティ
- **dcardmgr** : IPSec ドーターカードマネージャのロギングファシリティ
- **demuxmgr** : Demux Manager API ファシリティ
- **dgmbmgr** : Diameter Gmb アプリケーション マネージャのロギングファシリティ
- **dhcp** : Dynamic Host Configuration Protocol (DHCP) のロギングファシリティ
- **dhcpv6** : DHCPv6
- **dhost** : 分散ホストのロギングファシリティ
- **diabase** : Diabase メッセージファシリティ
- **diactrl** : Diameter Controller Proctlet のロギングファシリティ
- **diameter** : Diameter エンドポイントのロギングファシリティ
- **diameter-acct** : Diameter アカウンティング
- **diameter-auth** : Diameter 認証
- **diameter-dns** : Diameter DNS サブシステム
- **diameter-ecs** : ACS Diameter シグナリングファシリティ
- **diameter-engine** : Diameter バージョン 2 エンジンのロギングファシリティ
- **diameter-hdd** : Diameter Horizontal Directional Drilling (HDD) インターフェイスファシリティ
- **diameter-svc** : Diameter サービス
- **diamproxy** : DiamProxy のロギングファシリティ
- **doulosuemgr** : DOULOS (IMS-IPSec) ユーザー機器マネージャ
- **dpath** : IPSec データパスのロギングファシリティ

- **drvctrl** : ドライバコントローラのロギングファシリティ
- **eap-diameter** : Extensible Authentication Protocol (EAP; 拡張可能認証プロトコル) IP セキュリティファシリティ
- **eap-ipsec**: Extensible Authentication Protocol (EAP; 拡張可能認証プロトコル) IPSec ファシリティ
- **eap-sta-s6a-s13-s6b-diameter** : EAP/STA/S6A/S13/S6B Diameter メッセージファシリティ
- **ecs-css** : ACSMGR <-> セッション マネージャ シグナリング インターフェイス ファシリティ
- **egtpc** : eGTP-C のロギングファシリティ
- **egtpmgr** : 拡張 GPRS トンネリングプロトコル (eGTP) マネージャのロギングファシリティ
- **egtpu** : eGTP-U のロギングファシリティ
- **embms** : eMBMS ゲートウェイファシリティ
- **embms** : eMBMS ゲートウェイ Demux ファシリティ
- **epdg** : evolved Packet Data (ePDG) ゲートウェイのロギングファシリティ
- **event-notif** : イベント通知インターフェイスのロギングファシリティ
- **evlog** : イベントログファシリティ
- **famgr** : 外部エージェントマネージャのロギングファシリティ
- **firewall** : ファイアウォールのロギングファシリティ
- **fng** : フェムト ネットワーク ゲートウェイ (FNG) のロギングファシリティ
- **gbmgr** : SGSN Gb インターフェイス マネージャ ファシリティ
- **gmm** :
  - 2.5 G の場合 : GPRS Mobility Management (GMM) レイヤをログに記録します (LLC レイヤの上)
  - 3G の場合 : アクセス アプリケーション レイヤ (RANAP レイヤの上) をログに記録します。
- **gprs-app** : GPRS アプリケーションのロギングファシリティ
- **gprs-ns** : GPRS ネットワーク サービス プロトコル (SGSN と BSS の間のレイヤ) のロギングファシリティ
- **gq-rx-tx-diameter** : Gq/Rx/Tx Diameter メッセージファシリティ
- **gss-gcdr** : GTPP ストレージサーバーの GCDR ファシリティ

- **gtpc** : GTP-C プロトコルのロギングファシリティ
- **gtpcmgr** : GTP-C プロトコルマネージャのロギングファシリティ
- **gtpp** : GTP-prime プロトコルのロギングファシリティ
- **gtpu** : GTP-U プロトコルのロギングファシリティ
- **gtpumgr** : GTP-U Demux マネージャ
- **gx-ty-diameter** : Gx/Ty Diameter メッセージファシリティ
- **gy-diameter** : Gy Diameter メッセージファシリティ
- **h248prt** : H.248 ポートマネージャファシリティ
- **hamgr** : ホーム エージェント マネージャのロギングファシリティ
- **hat** : 高可用性タスク (HAT) プロセスファシリティ
- **hdctrl** : HD コントローラのロギングファシリティ
- **henbapp** : Home Evolved NodeB (HENB) アプリファシリティ
- **henbgw** : HENB-GW ファシリティ
- **henbgw-pws** : HENB-GW Public Warning System ロギングファシリティ
- **henbgw-sctp-acs** : HENB-GW アクセス Stream Control Transmission Protocol (SCTP) ファシリティ
- **henbgw-sctp-nw** : HENBGW ネットワーク SCTP ファシリティ
- **henbgwdemux** : HENB-GW Demux ファシリティ
- **henbgwmgr** : HENB-GW マネージャファシリティ
- **hnb-gw** : HNB-GW (3G Femto GW) ロギングファシリティ
- **hnbmgr** : HNB-GW Demux マネージャ ロギング ファシリティ
- **hss-peer-service** : ホームサブスクライバサーバー (HSS) ピアサービスファシリティ
- **iftask** : VPC-SI および VPC-DI プラットフォームで使用される内部フォワーダータスク (INTEL DPDK)
- **igmp** : Internet Group Management Protocol (IGMP)
- **ikev2** : インターネット キー エクスチェンジバージョン 2 (IKEv2)
- **ims-authorizatn** : IP マルチメディアサブシステム (IMS) 認証サービスファシリティ
- **ims-sh** : Diameter Sh インターフェイスサービスファシリティ
- **imsimgr** : SGSN IMSI マネージャファシリティ
- **imsue** : IMS User Equipment (IMSUE) ファシリティ

- **ip-arp** : IP Address Resolution Protocol (ARP)
- **ip-interface** : IP インターフェイスファシリティ
- **ip-route** : IP ルートファシリティ
- **ipms** : インテリジェント パケット モニタリング システム (IPMS) のロギングファシリティ
- **ipne**: IP Network ENABLER (IPNE) ファシリティ
- **ipsec** : IP セキュリティ ロギング ファシリティ
- **ipsecdemux** : IPSec demux のロギングファシリティ
- **ipsg** : IP サービス ゲートウェイ インターフェイスのロギングファシリティ
- **ipsgmgr**: IP サービス ゲートウェイ ファシリティ
- **ipsp** : IP プール共有プロトコルのロギングファシリティ
- **kvstore** : Key/Value ストア (kvstore) ファシリティ
- **l2tp-control** : レイヤ 2 トンネリングプロトコル (L2TP) コントロールのロギングファシリティ
- **l2tp-data** : L2TP データロギングファシリティ
- **l2tpdemux**: L2TP Demux マネージャのロギングファシリティ
- **l2tpmgr**: L2TP マネージャのロギングファシリティ
- **lagmgr** : Link Aggregation GROUP (LAG) マネージャのログインファシリティ
- **lcs** : ロケーションサービス (LCS) のロギングファシリティ
- **ldap** : ライトウェイトディレクトリ アクセス プロトコル
- **li** : コマンドの説明については、『*Lawful Intercept Configuration Guide*』を参照してください。
- **linkmgr** : SGSN/BSS SS7 リンクマネージャのロギングファシリティ (2.5G のみ)
- **llc** : 論理リンク制御 (LLC) プロトコルのロギングファシリティ。SGSN の場合、MS と SGSN の間の論理リンクについて、GMM レイヤと BSSGP レイヤの間にある LLC レイヤをログに記録します。
- **local-policy** : ローカル ポリシー サービス ファシリティ
- **location-service** : ロケーション サービス ファシリティ
- **m3ap** : M3 アプリケーションプロトコル ファシリティ
- **m3ua** : M3UA プロトコル ロギング ファシリティ
- **magmgr** : モバイル アクセス ゲートウェイ マネージャのロギングファシリティ

- **map** : モバイル アプリケーション パート (MAP) プロトコルのロギングファシリティ
- **megadiammgr** : MegaDiameter マネージャ (SLF サービス) のロギングファシリティ
- **mme-app** : モビリティ マネージメント エンティティ (MME) アプリケーションのロギングファシリティ
- **mme-embmsembms** : MME eMBMS ファシリティ
- **mme-misc** : MME その他のロギングファシリティ
- **mmedemux** : MME Demux マネージャのロギングファシリティ
- **mmemgr** : MME マネージャファシリティ
- **mmgr** : マスター マネージャのロギングファシリティ
- **mobile-ip** : モバイル IP プロセス
- **mobile-ip-data** : モバイル IP データファシリティ
- **mobile-ipv6** : モバイル IPv6 のロギングファシリティ
- **mpls** : マルチプロトコル ラベル スイッチング (MPLS) プロトコルのロギングファシリティ
- **mrme** : Multi Radio Mobility Entity (MRME) ロギングファシリティ
- **mseg-app** : Mobile Services Edge Gateway (MSEG) アプリケーション ロギング ファシリティ
- **mseg-gtpc** : MSEG GTP-C アプリケーション ロギング ファシリティ
- **mseg-gtpu** : MSEG GTP-U アプリケーション ロギング ファシリティ
- **msegmgr** : MSEG Demux マネージャ ロギング ファシリティ
- **mtp2** : Message Transfer Part 2 (MTP2) サービスのロギングファシリティ
- **mtp3** : Message Transfer Part 3 (MTP3) プロトコルのロギングファシリティ
- **multicast-proxy** : マルチキャストプロキシのロギングファシリティ
- **nas** : 非アクセス階層 (NAS) プロトコルのロギングファシリティ [MME 4G]
- **netwstrg** : ネットワーク ストレージファシリティ
- **npuctrl** : ネットワーク プロセッサ ユニット制御ファシリティ
- **npumgr** : ネットワーク プロセッサ ユニット マネージャ ファシリティ
- **npumgr-acl** : NPUMGR ACL のロギングファシリティ
- **npumgr-drv** : NPUMGR DRV のロギングファシリティ
- **npumgr-flow** : NPUMGR FLOW のロギングファシリティ



- **npumgr-fwd** : NPUMGR FWD のロギングファシリティ
- **npumgr-init** : NPUMGR INIT のロギングファシリティ
- **npumgr-lc** : NPUMGR LC のロギングファシリティ
- **npumgr-port** : NPUMGR PORT のロギングファシリティ
- **npumgr-recovery** : NPUMGR RECOVERY のロギングファシリティ
- **npumgr-rrri** : NPUMGR RRI (リバースルートインジェクション) のロギングファシリティ
- **npumgr-vpn** : NPUMGR VPN のロギングファシリティ
- **ocsp** : オンライン証明書ステータスプロトコル
- **orbs** : オブジェクトリクエストブローカ システムのロギングファシリティ
- **ospf** : OSPF プロトコルのロギングファシリティ
- **ospfv3** : OSPFv3 プロトコルのロギングファシリティ
- **p2p** : ピアツーピアの検出のロギングファシリティ
- **pagingmgr** : PAGINGMGR のロギングファシリティ
- **pccmgr** : Intelligent Policy Control Function (IPCF) Policy Charging and Control (PCC) マネージャライブラリ
- **pdg** : Packet Data Gateway (PDG) ロギングファシリティ
- **pdgdmgr** : Pdg Demux マネージャのロギングファシリティ
- **pdif** : Packet Data Interworking Function (PDIF) のロギングファシリティ
- **pgw** : Packet Data Network Gateway (PGW) ロギングファシリティ
- **pmm-app** : パケットモビリティ マネージメント (PMM) アプリケーションのロギングファシリティ
- **ppp** : ポイントツーポイントプロトコル (PPP) リンクとパケットファシリティ
- **pppoe** : PPP over Ethernet のロギングファシリティ
- **procllet-map-frwk** : Procllet マッピングフレームワークのロギングファシリティ
- **push** : VPNMGR CDR プッシュのロギングファシリティ
- **radius-acct** : RADIUS アカウンティング ロギング ファシリティ
- **radius-auth** : RADIUS 認証のロギングファシリティ
- **radius-coa** : RADIUS の認可変更および radius 接続解除
- **ranap** : Radio Access Network Application Part (RANAP) プロトコルの SGSN と RNS (3G) 間のファシリティロギング情報フロー

- **rc** : リカバリ制御タスクのロギングファシリティ
- **rd** : リダイレクトタスクのロギングファシリティ
- **resmgr** : Resource Manager のロギングファシリティ
- **rf-diameter** : Diameter Rf インターフェイス メッセージファシリティ
- **rip** : Routing Information Protocol (RIP) のロギングファシリティ [RIP は現時点ではサポートされていません。]
- **rlf** : レート制限機能 (RLF) のロギングファシリティ
- **rohc** : Robust Header Compression (RoHC) ファシリティ
- **rsvp** : 予約プロトコルのロギングファシリティ
- **rua** : RANAP ユーザーアダプテーション (RUA) [3G フェムト GW メッセージ] のロギングファシリティ
- **s102** : S102 プロトコルのロギングファシリティ
- **s102mgr** : S102Mgr のロギングファシリティ
- **s1ap** : S1 アプリケーションプロトコル (S1AP) プロトコルのロギングファシリティ
- **sabp** : Service Area Broadcast Protocol (SABP) のロギングファシリティ
- **saegw** : System Architecture Evolution (SAE) ゲートウェイファシリティ
- **sbc** : SBc プロトコルのロギングファシリティ
- **sccp** : Signalling Connection Control Part (SCCP) プロトコルのロギング (RANAP と TCAP レイヤ間のコネクション型メッセージ)。
- **sct** : 共有設定タスクのロギングファシリティ
- **sctp** : Stream Control Transmission Protocol (SCTP) プロトコルのロギングファシリティ
- **sef\_ecs** : Severely Errored Frames (SEF) API 印刷ファシリティ
- **sess-gr** : SM GR ファシリティ
- **sessctrl** : セッションコントローラのロギングファシリティ
- **sessmgr** : セッションマネージャのロギングファシリティ
- **sesstrc** : セッショントレースのロギングファシリティ
- **sft** : スイッチファブリックタスクのロギングファシリティ
- **sgs** : SG インターフェイスプロトコルのロギングファシリティ
- **sgsn-app** : SGSN-APP がさまざまな SGSN の「グルー」インターフェイスをロギング (たとえば、PMM、MAP、GPRS FSM、SMS など)。

- **sgsn-failures** : SGSN コールの失敗 (接続/アクティブ化の拒否) のロギングファシリティ (2.5G)
- **sgsn-gtpe** : SGSN と GGSN 間の制御メッセージをロギングする SGSN GTP-C プロトコル
- **sgsn-gtpu** : ユーザーデータメッセージをロギングする SGSN GTP-U プロトコル
- **sgsn-mbms-bearer** : SGSN マルチメディアブロードキャスト/マルチキャストサービス (MBMS) ベアラー APP (SMGR) のロギングファシリティ
- **sgsn-misc** : スタックマネージャが、レイヤ間のバインディングと削除をロギングするために使用する
- **sgsn-system** : SGSN システムコンポーネントのロギングファシリティ (使用頻度が低い)
- **sgsn-test** : SGSN テストのロギングファシリティ (使用頻度の低い)
- **sgtpcmgr** : SGTPC および GGSN を介した SGSN GTP-C Manager のロギング情報の交換
- **sgw** : サービング ゲートウェイ ファシリティ
- **sh-diameter** : Sh Diameter メッセージファシリティ
- **sitmain** : システム初期化タスクのメインロギングファシリティ
- [**slmgr.vbs**] : スマート ライセンシング マネージャのロギングファシリティ
- **sls** : サービスレベル仕様 (SLS) プロトコルのロギングファシリティ
- **sm-app** : SM プロトコルのロギングファシリティ
- **sms** : ショートメッセージサービス (SMS) による MS と SMSC の間のメッセージのロギング
- **sndcp** : Sub Network Dependent Convergence Protocol (SNDCP) のロギングファシリティ
- **snmp** : SNMP のロギングファシリティ
- **sprmgr** : IPCF Subscriber Policy Register (SPR) マネージャのロギングファシリティ
- **srdp** : スタティック評価データベース
- **srp** : サービス冗長性プロトコル (SRP) のロギングファシリティ
- **sscfnni** : Service-Specific Coordination Function for Signaling at the Network Node Interface (SSCF-NNI) のロギングファシリティ
- **sscop** : Service-Specific Connection-Oriented Protocol (SSCOP) のロギングファシリティ
- **ssh-ipsec** : Secure SHELL (SSH) IP セキュリティのロギングファシリティ
- **ssl** : Secure Socket Layer (SSL) メッセージのロギングファシリティ
- **stat** : 統計情報のロギングファシリティ
- **supserv** : 補足サービスのロギングファシリティ [H.323]

- **system** : システムのログGINGファシリティ
- **tacacsplus** : TACACS+ プロトコル ログGING ファシリティ
- **tcap** : TCAP プロトコルのログGINGファシリティ
- **testctrl** : テストコントローラのログGINGファシリティ
- **testmgr** : テストマネージャのログGINGファシリティ
- **threshold** : しきい値のログGINGファシリティ
- **ttg** : Tunnel Termination Gateway (TTG) のログGINGファシリティ
- **tucl** : TCP/UDP コンバージェンスレイヤ (TUCL) のログGINGファシリティ
- **udr** : ユーザーデータレコード (UDR) ファシリティ (Charging サービスで使用)
- **user-data** : ユーザーデータのログGINGファシリティ
- **user-l3tunnel** : ユーザーレイヤ3 トンネルのログGINGファシリティ
- **usertcp-stack** : ユーザー TCP スタック
- **vim** : Voice インスタントメッセージ (VIM) のログGINGファシリティ
- **vinfo** : VINFO のログGINGファシリティ
- **vmgctrl** : 仮想メディアゲートウェイ (VMG) コントローラファシリティ
- **vmgctrl** : VMG コンテンツ マネージャ ファシリティ
- **vpn** : 仮想プライベートネットワークのログGINGファシリティ
- **vpp** : Vector Packet Processing (VPP) のログGINGファシリティ
- **wimax-data** : WiMAX データ
- **wimax-r6** : WiMAX R6
- **wsg** : ワイヤレス セキュリティ ゲートウェイ (ASR 9000 セキュリティゲートウェイ)
- **x2gw-app** : X2GW (X2 プロキシゲートウェイ、eNodeB) アプリケーションのログGINGファシリティ
- **x2gw-demux** : X2GW demux タスクのログGINGファシリティ

## トレースログGINGの設定

トレースログGINGは、現在アクティブな特定のセッションの問題を迅速に解決するのに役立ちます。これらは、Exec モードで **logging filter** コマンドを使用して設定されたグローバル イベントログフィルタに依存しない修飾子に基づいて生成される一時的なフィルタです。ただし、

イベントログと同様に、ログによって生成される情報は、アクティブなメモリバッファに保存されます。

選択したコールに関連付けられているすべてのデバッグレベルイベントが保存されます。



**重要** トレースログは、セッションの処理に影響します。デバッグ目的でのみ実装する必要があります。

Exec モードでトレースログを設定するには、次の例を使用します。

```
[local]host_name# logging trace { callid call_id | ipaddr ip_address | msid  
ms_id | username username }
```

必要なすべての情報が収集されたら、次のコマンドを入力して、トレースログを削除できます。

```
[local]host_name# no logging trace { callid call_id | ipaddr ip_address | msid  
ms_id | username username }
```

## モニターログの設定

モニターロギングは、特定のサブスクリバのすべてのセッションに関連付けられているすべてのアクティビティを記録します。この機能は、特定のサブスクリバのモニターリング機能に関する司法当局の要請を遵守している場合に使用できます。

モニターは、サブスクリバのMSIDまたはユーザー名に基づいて実行できます。また、司法当局によって規定された限定期間のみを使用することを目的としています。したがって、必要なモニターリング期間の直後に終了する必要があります。

この項では、モニターログを有効または無効にする手順について説明します。

## モニターログの有効化

モニターログのターゲットを設定するには、次の例を使用します。

```
configure  
logging monitor { ip_addr | ipv6_addr | msid id | username name }  
end
```

モニターログの追加のターゲットを設定するには、この手順を繰り返します。

## モニターログの無効化

モニターログを無効にするには、次の例を使用します。

```
configure  
no logging monitor { ip_addr | ipv6_addr | msid id | username name }  
end
```

## ロギング設定と統計情報の表示

ロギング設定と統計情報を確認するには、Exec モードで次のコマンドを入力します。

```
[local]host_name# show logging [ active | verbose ]
```

キーワードを指定しなかった場合は、グローバルフィルタ設定が表示され、有効になっている他のタイプのロギングに関する情報も表示されます。

次の表に、**verbose** キーワードを使用したときに表示される統計情報の説明を示します。

表 13: ロギング設定コマンドと統計情報コマンド

フィールド	説明
<b>General Logging Statistics</b>	
Total events received	システムによって生成されたイベントの合計数が表示されます。
Number of applications receiving events	イベントを受信しているアプリケーションの数が表示されます。
<b>Logging Source Statistics</b>	
Event sequence ids by process	生成されたイベントがあるシステムプロセスのリストと、生成されたイベントの参照識別番号が表示されます。
Msg backlog stat with total cnt	生成されたイベントの合計数と比較してログに記録されたイベントメッセージの数が表示されます。
LS L2 filter drop rate	ロギングソース (LS) のレイヤ2 (L2) イベントのドロップのパーセンテージが表示されます。
Abnormal Log Source Statistics	異常なロギングソース (LS) の統計情報が表示されます (存在する場合)。
<b>Runtime Logging Buffer Statistics</b>	
Active buffer	現在アクティブなメモリバッファに記録されているイベントの数と、バッファ内の最も古いエントリと最新のエントリのタイムスタンプが表示されます。
Inactive buffer	非アクティブなメモリバッファに現在ログインしているイベントの数が表示されます。

## CLI を使用したイベントログの表示

システムによって生成されるイベントログは、次のいずれかの方法で表示できます。

- **syslog** サーバーから、次のようにします。システムが syslog サーバーにログを送信するように設定されている場合、ログは syslog サーバーで直接表示できます。

- **システム CLI から、次のようにします。** システムのメモリバッファに保存されているログは、CLI から直接表示できます。
- **コンソールポートから、次のようにします。** デフォルトでは、CLI セッションがアクティブになっていない場合、システムはコンソールインターフェイスを介して端末にイベントを自動的に表示します。

この項では、CLI を使用してイベントログを表示する手順を説明します。これらの手順は、Exec モードのルートプロンプトを使用していることが前提になります。

**ステップ 1** アクティブなログメモリバッファを非アクティブなログメモリバッファにコピーします。

アクティブなログメモリバッファが非アクティブなログメモリバッファにコピーされると、非アクティブなログメモリバッファ内の既存の情報が削除されます。

アクティブと非アクティブの両方のイベントログメモリバッファは、Exec モードの CLI を使用して表示できます。ただし、データが上書きされないようにするために、非アクティブなログを表示することをお勧めします。アクティブなログバッファからの情報は、次のコマンドを入力して非アクティブなログバッファにコピーできます。

```
[local]host_name# logs checkpoint
```

**ステップ 2** 次のコマンドを入力してログを表示します。

```
[local]host_name# show logs
```

**show logs** コマンドでは、いくつかのオプションのキーワードや変数を使用できます。詳細については、『*Command Line Interface Reference*』にある「*Exec Mode Show Commands*」の章を参照してください。

## クラッシュログの設定と表示

ソフトウェアのクラッシュが発生した場合は、クラッシュの原因を特定するのに役立つ情報をシステムが保存します。この情報はシステムメモリ内で維持することも、ネットワークサーバーに転送して保存することもできます。

システムでは、次の 2 種類のログを生成することができます。

- **クラッシュログ** : クラッシュログには、ソフトウェアクラッシュに関するあらゆる情報が記録されます (完全なコアダンプ)。サイズが原因で、システムメモリに保存することはできません。したがって、ログを保存できるローカルデバイスか、またはネットワークサーバーを指定する Universal Resource Locator (URL) を使用して設定されている場合のみ、これらのログが生成されます。
- **クラッシュログの要約** : クラッシュイベントレコードは、ソフトウェアクラッシュが発生したときに自動的に生成され、管理カード上のフラッシュメモリに保存されます。クラッシュログの要約には、関連付けられたダンプファイルとともに、クラッシュイベントのレ

コードのリストが含まれています。このログは、CLI コマンドを使用してイベントレコードとダンプファイルを表示します。

## クラッシュロギングのアーキテクチャ

クラッシュログは、クラッシュイベント情報の永続的なリポジトリです。各イベントには番号が付けられており、CPU (minicore)、NPU、またはカーネルクラッシュに関連するテキストが含まれます。ログに記録されたイベントは、固定長レコードに記録され、`/flash/crashlog2` に保存されます。

クラッシュが発生するたびに、次のクラッシュ情報が保存されます。

1. イベントレコードは、`/flash/crashlog2` ファイル (クラッシュログ) に保存されます。
2. 関連する minicore、NPU、またはカーネルダンプファイルは、`/flash/crsh2` ディレクトリに保存されます。
3. 完全なコアダンプは、ユーザーが設定したディレクトリに保存されます。



---

**重要** `crashlog2` ファイルは、関連する minicore、NPU、およびカーネルダンプとともに、冗長管理カード (SMC、MIO/UMIO) 間で自動的に同期されます。フルコアダンプは、管理カード間で同期されません。

---

次の動作は、クラッシュロギングプロセスに適用されます。

- クラッシュイベントがアクティブな管理カードに到着すると、イベントレコードは、`/flash/crsh2` の minicore、NPU、またはカーネルダンプファイルとともに `crashlog2` ファイルに保存されます。クラッシュイベントとダンプファイルも、スタンバイ管理カード上の同じ場所に自動的に保存されます。
- クラッシュログエントリが CLI コマンドを使用して削除されると、アクティブとスタンバイの両方の管理カードで削除されます。
- 管理カードを追加または交換すると、アクティブカードとスタンバイカードによってクラッシュログとダンプファイルが自動的に同期されます。
- クラッシュイベントが受信され、クラッシュログファイルがいっぱいになると、クラッシュログ内の最も古いエントリと関連するダンプファイルが、両方の管理カードの最新の到着イベントとダンプファイルに置き換えられます。最大 120 のクラッシュイベントの情報を、管理カードに保存できます。
- クラッシュイベントが重複すると、既存のレコードのヒット数が変更され、古いクラッシュレコードで新しいレコードが更新されます。カウントに追加すると、イベントが最初に発生したときにタイムスタンプが使用されます。



## ソフトウェアクラッシュログ接続先の設定

システムは、ソフトウェアのクラッシュログ情報を次のいずれかの場所に保存するように設定できます。

- ASR 5500 上：
  - **フラッシュメモリ**：アクティブな MIO/UMIO に取り付けられています（要約されたクラッシュログおよび関連するダンプファイルのみ）。
  - **USB メモリスティック**：アクティブな MIO/UMIO の USB スロットに取り付けられています。
- VPC 上
  - **フラッシュメモリ**：仮想マシンからアクセス可能です。
  - **USB メモリスティック**：プラットフォームの USB スロットに取り付けられています（USB スロットはハイパーバイザを介して有効になっています）。
- **ネットワークサーバー**：システムが Trivial File Transfer Protocol (TFTP)、File Transfer Protocol (FTP)、Secure File Transfer Protocol (SFTP)、または Hypertext Transfer Protocol (HTTP) を使用してアクセスできるネットワーク上のワークステーションまたはサーバーです。これは、複数のシステムが同じ設定を必要とする大規模なネットワークの展開に推奨されます。



**重要** FTP はサポートされていません。

クラッシュログファイル（完全なコアダンプ）は、指定された場所で発生すると一意の名前で書き込まれます。名前の形式は、*crash-card-cpu-time-core* です。*Card* はカードスロット、*cpu* はカード上の CPU の数、*time* は 16 進表記の Portable Operating System Interface (POSIX) のタイムスタンプです。

次の例を使用して、グローバル コンフィギュレーション モードでソフトウェアのクラッシュログの接続先を設定します。

### configure

```
crash enable [ encrypted ] url crash_url
end
```

注：

- このコマンドの詳細については、『*Command Line Interface Reference*』の「*Global Configuration Mode Commands*」の章を参照してください。
- 追加のソフトウェアのクラッシュログの接続先を設定するには、この手順を繰り返します。設定可能な接続先の数に制限はありません。

「設定の確認と保存」の章の説明に従って、設定を保存します。

## CLI を使用して要約されたクラッシュログ情報の表示

管理カード (**/flash/crashlog2**) 上のフラッシュメモリに一連のイベントレコードとして保存されている要約されたクラッシュ情報を表示できます。各クラッシュイベントレコードには、表示可能な (minicore、NPU、またはカーネル) 関連のダンプファイル (**/flash/crsh2**) があります。

システムで発生したソフトウェアクラッシュイベントを表示するには、この項の手順に従ってください。これらの手順は、Exec モードのルートプロンプトを使用していることが前提になります。

**ステップ 1** 次の Exec モードコマンドを入力して、ソフトウェアクラッシュイベントのリストを表示します。

```
[local]host_name# show crash { all | list | number crash_num }
```

注：

- **Show crash list** を実行して、特定のクラッシュイベントの番号を取得します。
- **Show crash number crash\_num** を実行して、ターゲットクラッシュイベントの出力を表示します。

結果として得られる出力は、すべてのプラットフォームで同じとは限りません。

同様のクラッシュイベントに関する情報は、このコマンドの出力では抑制されます。

**ステップ 2** 特定のクラッシュイベントに関連付けられているダンプファイルを表示します。

ダンプファイルに含まれる情報は、ソフトウェアがクラッシュする原因となっている内部または外部の要因を特定して診断するのに役に立ちます。

- クラッシュ #：クラッシュイベントをログに記録するときに StarOS によって割り当てられた一意の番号
- SW バージョン：StarOS ビルドリリース形式：RR.n(bbbbb)
- 同様のクラッシュカウント：類似したクラッシュの数
- 最初のクラッシュ時刻：YYYY-MMM-DD+hh:mm:ss の形式で最初のクラッシュが発生したときのタイムスタンプ
- 失敗メッセージ：イベントメッセージのテキスト
- 関数：コード識別子
- プロセス：クラッシュが発生した場所 (カード、CPU、PID など)
- クラッシュ時間：クラッシュが発生したときのタイムスタンプ (YYYY-MMM-DD+hh:mm:ss タイムゾーン)
- 最近の **errno**：最新のエラー番号のテキスト。
- スタック：メモリスタック情報
- ラストバウンズ：クラッシュ前に受信したメッセージングに関する情報
- レジスタ：メモリレジスタの内容
- 現在の着信メッセージ：現在の着信メッセージの 16 進情報
- アドレス マップ
- 最近のヒープアクティビティ (最も古いもの)
- 最近のイベント (最も古いもの)

- プロファイルの深さ

各クラッシュログエントリの情報内容は、クラッシュのタイプと StarOS リリースによって異なります。

## 過剰なイベントロギングの削減

イベントロギング (evlogd) は、StarOS ファシリティによって送信されたイベントメッセージをキャプチャする共有メディアです。1 つまたは複数のファシリティが継続的に大量のイベントメッセージを送信し続けると、残りの通常に動作している機能が影響を受けます。このシナリオでは、特にログを生成するファシリティの数が増えるにつれて、システムパフォーマンスが低下していきます。

イベントメッセージロギングのレートコントロールは、ログの送信元パスで処理されます。基本的に、カウンタは秒ごとに 0 に設定され、evlogd に送信されたログイベントごとに増分します。1 秒未満でカウンタがしきい値に達すると、イベントは送信され、キューに入るか、または (evlogd メッセージキューがいっぱいの場合) ドロップされます。

メッセージロギングのレートに対するこのコマンドで設定された上限しきい値を超過するファシリティがあり、同じ状態が長時間続く場合、StarOS は SNMP トラップまたはアラームを介してユーザーに通知します。

新しい threshold コマンドを使用すると、ユーザーはファシリティイベントキューを満杯とするパーセンテージを指定できます。このしきい値を超えると、問題のあるファシリティを指定する SNMP トラップとアラームが生成されます。

このコマンドに関連付けられている SNMP トラップの形式は次のとおりです。

### • ThreshLSLogsVolume

```
<timestamp> Internal trap notification <trap_id> (ThreshLSLogsVolume) threshold  
<upper_percent>%  
measured value <actual_percent>% for facility <facility_name> instance <instance_id>
```

### • ThreshClearLSLogsVolume

```
<timestamp> Internal trap notification <trap_id> (ThreshClearLSLogsVolume) threshold  
<upper_percent>%  
measured value <actual_percent>% for facility <facility_name> instance <instance_id>
```

ポーリング間隔内にトリガー条件が発生した場合は、そのポーリング間隔が終了するまではアラートやアラームは生成されません。

両方のトラップを有効または抑制するには、グローバル コンフィギュレーション モードの **snmp trap** コマンドを使用します。

## ログソースのしきい値の設定

ログソースのしきい値の設定と実装に関連する 3 つのグローバル コンフィギュレーション モードコマンドがあります。

1. **threshold ls-logs-volume** : トラップやアラームをそれぞれ生成およびクリアするための上限および下限しきい値のパラメータを設定します。
2. **threshold poll ls-logs-volume interval** : このしきい値のポーリング間隔を設定します。
3. **threshold monitoring ls-logs-volume** : このしきい値のモニタリングをオンまたはオフにします。

Syslog サーバーを設定するには、次の例を使用します。

**configure**

```
[ default ] threshold ls-logs-volume upper_percent [ clear lower_percent ]
[ default ] threshold poll ls-logs-volume interval duration
[ no ] threshold monitoring ls-logs-volume
end
```

注 :

- *upper\_percent* と *lower\_percent* は、0 ~ 100 の整数として表わされます。 *upper\_percent* のデフォルト値は 90% です。 *lower\_percent* が指定されていない場合、デフォルトのクリア値は *upper\_percent* です。
- **threshold poll ls-logs-volume interval** はポーリング間隔を秒単位で設定します。デフォルトの間隔は 300 秒 (5 分) です。
- **threshold monitoring ls-logs-volume** はこの機能を有効または無効にします。

このしきい値の設定を確認するには、Exec モードの **show threshold** コマンドを実行します。

「設定の確認と保存」の章の説明に従って、設定を保存します。

## ログのチェックポイントニング

チェックポイントニングは、ログに記録されたデータが以前に表示されたものか、マークされたものかを識別します。チェックポイントニングを使用すると、最後のチェックポイント以降のログ情報のみを表示できます。

個々のログには、アクティブログに最大 50,000 のイベントが含まれている場合があります。ログのチェックポイントを実行すると、最大 50,000 のイベントが非アクティブなログファイルに記録されます。これにより合計で最大 100,000 イベントとなり、これらはログに記録された各ファシリティで使用できます。

Exec モードの **logs checkpoint** コマンドを使用してログデータのチェックポイントニングを実行し、特別なアクティビティが発生する前にログの内容を既知のポイントに設定します。また、このコマンドを定期メンテナンスの一環として含めて、ログデータを管理することもできます。

ログのチェックポイントニングでは、現在のログデータを非アクティブなログに移動します。最後にチェックポイントニングされたデータが、非アクティブログに保持されます。後続のログのチェックポイントニングによって、以前にチェックポイントニングされた非アクティブな

ログデータがクリアされ、新たにチェックポイントされたデータに置き換えられます。チェックポイントされたログデータは表示できません。



**重要** ログファイルがいっぱいになるのを防ぐために、ログのチェックポイントを定期的に行う必要があります。50,000のイベントが記録されたログは、新しいイベントがログに記録されると、最も古いイベントを最初に破棄します。



**重要** インспекタレベルの管理ユーザーは、このコマンドを実行できません。

## ログファイルの保存

ログファイルは、URL で指定されたローカルまたはリモートの場所にあるファイルに保存できます。ログファイルを保存するには、次の Exec モードコマンドを使用します。

```
save logs { url } [ active ] [ inactive ] [ callid call_id ]
[ event-verbosity evt_verbosity ] [ facility facility ] [ level severity_level ]
[ pdu-data pdu_format ] [ pdu-verbosity pdu_verbosity ] [ since
from_date_time [ until to_date_time ] ] [ | { grep grep_options | more } ]
```

**save logs** コマンドの詳細については、『*Command Line Interface Reference*』の「*Exec Mode Commands*」の章を参照してください。

## イベント ID の概要



**重要** イベント ID の使用は、プラットフォームのタイプとプラットフォームで実行されているライセンスによって異なります。

識別番号 (ID) は、システムでロギングが有効になっている場合に発生するイベントを参照するために使用されます。前述したように、ログはファシリティごとに収集されます。各ファシリティには、次の表に示すように、独自の範囲のイベント ID があります。

表 14: システムファシリティとイベント ID の範囲

ファシリティ	説明	イベント ID の範囲
a10	A10 プロトコルファシリティ	28000 ~ 28999
a11	A11 プロトコルファシリティ	29000 ~ 29999
allmgr	A11 マネージャファシリティ	9000 ~ 9999

ファシリティ	説明	イベント ID の範
aaa-client	AAA クライアントファシリティ	6000 ~ 6999
aaamgr	AAA マネージャファシリティ	36000 ~ 36999
aaaproxy	AAA プロキシファシリティ	64000 ~ 64999
aal2	AAL2 プロトコルファシリティ	173200 ~ 173299
acl-log	IP アクセスコントロールリスト (ACL) ファシリティ	21000 ~ 21999
acsctrl	アクティブチャージングサービスコントローラ (ACSCtrl) ファシリティ	90000 ~ 90999
acsmgr	アクティブチャージングサービスマネージャ (ACSMgr) ファシリティ	91000 ~ 91999
afctrl	Ares ファブリックコントローラ (ASR 5500 のみ)	186000 ~ 186999
afmgr	Ares Fabric Manager (ASR 5500 のみ)	187000 ~ 187999
alarmctrl	アラームコントローラファシリティ	65000 ~ 65999
alcap	Access Link Control Application Part (ALCAP) プロトコルファシリティ	160900 ~ 161399
alcapmgr	ALCAP マネージャファシリティ	160500 ~ 160899
asf	ASF ファシリティ	73000 ~ 73999
asfprt	ASFPRT ファシリティ	59000 ~ 59999
asngwmgr	アクセスサービスネットワーク (ASN) ゲートウェイ マネージャファシリティ	100000 ~ 100499
asnpcmgr	ASN ページング/ロケーションレジストリ マネージャファシリティ	100500 ~ 100999
bcmcs	ブロードキャスト/マルチキャストサービス (BCMCS) ファシリティ	109000 ~ 109999
bfd	Bidirectional Forwarding Detection (BFD) プロトコルファシリティ	170500 ~ 170999
bgp	ボーダーゲートウェイプロトコル (BGP) ファシリティ	85000 ~ 85999
bindmux	BindMux マネージャファシリティ [インテリジェントポリシー制御機能 (IPCF) ]	158200 ~ 158999
bngmgr	ブロードバンドネットワークゲートウェイ (BNG) マネージャファシリティ	182000 ~ 182999
bssap	Base Station System Application Part+ (BSSAP+) サービスファシリティ	131000 ~ 131199
bssgp	Base Station System GPRS Protocol (BSSGP) ファシリティ	115050 ~ 115099
callhome	Call Home ファシリティ	173600 ~ 173999

ファシリティ	説明	イベント ID
cap	CAMEL Application Part (CAP) ファシリティ	87900 ~ 88000
chatconf	CHATCONF ファシリティ	74000 ~ 74999
cli	CLI (コマンドラインインターフェイス) のログインファシリティ	30000 ~ 30999
connproxy	接続プロキシファシリティ	190000 ~ 199999
crdt-ctl	クレジット制御ファシリティ	127000 ~ 127999
csg	Closed Subscriber Groups (CSG) ファシリティ	188000 ~ 188999
csg-acl	CSG アクセスコントロールリスト (ACL) ファシリティ	189000 ~ 189999
csp	カード/スロット/ポート (CSP) ファシリティ	7000 ~ 7999
css	コンテンツ ステアリング サービス (CSS) ファシリティ [ESC]	77000 ~ 77499
css-sig	コンテンツ サービス セレクション (CSS) RADIUS シグナリング ファシリティ	77500 ~ 77599
cx-diameter	Cx Diameter メッセージファシリティ	92840 ~ 92899
dcardctrl	ドーターカードコントローラファシリティ	62000 ~ 62999
dcardmgr	ドーターカード マネージャ ファシリティ	57000 ~ 57999
demuxmgr	Demux マネージャファシリティ	110000 ~ 110999
dgmbmgr	Diameter Gmb (DGMB) アプリケーション マネージャ ファシリティ	126000 - 126999
dhcp	DHCP ファシリティ	53000 ~ 53999
dhcpv6	DHCPv6 プロトコルファシリティ	123000 ~ 123999
dhost	分散型ホスト マネージャ ファシリティ	83000 ~ 83999
diameter	Diameter エンドポイントファシリティ	92000 ~ 92599
diabase	Diabase メッセージファシリティ	92800 ~ 92899
diameter-acct	Diameter アカウンティング プロトコル ファシリティ	112000 ~ 112999
diameter-auth	Diameter 認証プロトコルファシリティ	111000 ~ 111999
diameter-dns	Diameter DNS サブシステムファシリティ	92600 ~ 92699
diameter-ecs	ECS Diameter シグナリングファシリティ	81990 ~ 81999
diameter-hdd	Diameter Horizontal Directional Drilling (HDD) インターフェイス ファシリティ	92700-92799
diameter-svc	Diameter サービスファシリティ	121200 ~ 121299
diamproxy	Diameter プロキシファシリティ	119000 ~ 119999
dpath	IPSec ファシリティのデータパス	54000 ~ 54999

ファシリティ	説明	イベント ID の範
drvctrl	ドライバコントローラ ファシリティ	39000 ~ 39999
ds3mgr	DS3 および DS3/E ラインカード マネージャ ファシリティ (NPU マネージャ コントローラ ファシリティの一部)	40000 ~ 40999
eap-diameter	Extensible Authentication Protocol (EAP; 拡張可能認証プロトコル) Diameter ファシリティ	92870 ~ 92879
eap-ipsec	EAP IPSec ファシリティ	118000 ~ 118999
ecs-css	ACSセッションマネージャ (ACSMgr) シグナリングインターフェイス ファシリティ	97000 ~ 97099
edr	イベントデータレコード (EDR) ファシリティ	80000 ~ 80999
egtpc	eGTP-C ファシリティ	141000 ~ 141999
egtpmgr	eGTP マネージャファシリティ	143000 ~ 143999
egtpu	eGTP-U ファシリティ	142000 ~ 142999
epdg	Evolved Packet Data Gateway (ePDG) ファシリティ	178000 ~ 178999
evlog	イベントログファシリティ	2000 ~ 2999
famgr	外部エージェント (FA) マネージャファシリティ	33000 ~ 33999
ファイアウォール	ファイアウォール機能	96000 ~ 96999
fng	フェムト ネットワーク ゲートウェイ (FNG) ファシリティ	149000 ~ 149999
gbrmgr	Gb-Manager ファシリティ	201900 ~ 202699
gcdr	GGSN 課金データレコード (G-CDR) ファシリティ	66000 ~ 66999
gmm	GPRS Mobility Management (GMM) ファシリティ	88100 ~ 88299
gprs-app	General Packet Radio Service (GPRS) アプリケーション ファシリティ	115100 ~ 115399
gprs-ns	GPRS-NS プロトコルファシリティ	115000 ~ 115049
gq-rx-tx-diameter	Gq/Rx/Tx Diameter メッセージファシリティ	92830 ~ 92839
gss-gcdr	GTPP ストレージサーバーの GCDR ファシリティ	98000 ~ 98099
gtpc	GTPC プロトコルファシリティ	47000 ~ 47999
gtpcmgr	GTPC シグナリング デマルチプレクサ マネージャ ファシリティ	46000 ~ 46999
gtp	GTP-PRIME プロトコルファシリティ	52000 ~ 52999
gtpu	GTPU プロトコルファシリティ	45000 ~ 45999
gtpumgr	GTPU マネージャ ファシリティ	157200 ~ 157999
gx-ty-diameter	Gx/Ty Diameter メッセージファシリティ	92820 ~ 92829



ファシリティ	説明	イベント ID の概要
gy-diameter	Gy Diameter メッセージファシリティ	92810 ~ 928
h248prt	H.248 プロトコルファシリティ	42000 - 42999
hamgr	ホームエージェント (HA) マネージャファシリティ	34000 ~ 349
hat	高可用性タスク (HAT) ファシリティ	3000 ~ 3999
hdctrl	ハードディスク (HD) コントローラファシリティ	132000 ~ 13
hddshare	HDD 共有ファシリティ	184000 ~ 18
henb-gw	Home eNodeB-GW ファシリティ	195000 ~ 19
henbapp	Home eNodeB アプリケーションファシリティ	196000 ~ 19
henbgwdemux	Home eNodeB-GW Demux ファシリティ	194000 ~ 19
henbgwmgr	Home eNodeB-GW マネージャファシリティ	193000、193
hnb-gw	Home NodeB (HNB) ゲートウェイファシリティ	151000 ~ 15
hnbmgr	HNB マネージャファシリティ	158000 ~ 15
hss-peer-service	ホーム サブスクライバ サーバー (HSS) ファシリティ [MME]	138000 ~ 13
igmp	Internet Group Management Protocol (IGMP)	113000 ~ 11
ikev2	IKEv2 ファシリティ	122000 ~ 12
ims-authorization	IMS 承認サービス ライブラリ ファシリティ	98100 ~ 989
ims-sh	IMS SH ライブラリファシリティ	124000 ~ 12
imsimgr	International Mobile Subscriber Identity (IMSI) マネージャファシリティ	114000 ~ 11
imsue	IMS User Equipment (IMSUE) ファシリティ	144000 ~ 14
ip-arp	IP Address Resolution Protocol (ARP) ファシリティ	19000 ~ 199
ip-interface	IP インターフェイスファシリティ	18000 ~ 189
ip-route	IP ルートファシリティ	20000 ~ 209
ipms	インテリジェントパケットモニタリングシステム (IPMS) ファシリティ	134000 ~ 13
ipne	IP Network ENABLER (IPNE) ファシリティ	192000 ~ 19
ipsec	IPSec プロトコルファシリティ	55000 ~ 569
ipsg	IP サービスゲートウェイ (IPSG) ファシリティ	128000 ~ 12
ipsgmgr	IPSG マネージャ (IPSGMgr) ファシリティ	99000 ~ 999
ipsp	IP プール共有プロトコル (IPSP) ファシリティ	68000 ~ 689

ファシリティ	説明	イベント ID の範
kvstore	Key/Value ストア (kvstore) ファシリティ	125000 ~ 125999
l2tp-control	L2TP Control PDU Protocol ファシリティ	50000 ~ 50999
l2tp-data	L2TP Data PDU Protocol ファシリティ	49000 ~ 49999
l2tpdemux	L2TP Demux ファシリティ	63000 ~ 63999
l2tpmgr	L2TP マネージャファシリティ	48000 ~ 48999
lagmgr	Link Aggregation GROUP (LAG) マネージャファシリティ	179000 ~ 179999
ldap	Lightweight Directory Access Protocol (LDAP) リクエストファシリティ	160000 ~ 160499
li	合法的傍受 (LI) ログファシリティ	69000 ~ 69999
linkmgr	リンク マネージャ ファシリティ	89500 ~ 89999
llc	論理リンク制御 (LLC) レイヤファシリティ (GPRS)	115700 ~ 115799
local-policy	ローカルポリシー設定ファシリティ	161400 ~ 162399
m3ap	M3 アプリケーションプロトコル (M3AP) ファシリティ	211500 ~ 211999
m3ua	MTP レベル 3 (M3UA) プロトコルファシリティ [SIGTRAN]	87500 ~ 87699
magmgr	モバイルアクセスゲートウェイ (MAG) マネージャファシリティ	137500 ~ 137999
map	モバイルアプリケーションパート (MAP) プロトコルファシリティ [SS7]	87100 ~ 87299
megadiammgr	MegaDiameter マネージャファシリティ	121000 ~ 121199
mme-app	モビリティマネージメントエンティティ (MME) アプリケーションファシリティ	147000 ~ 147999
mme-embms	MME evolved Multimedia Broadcast Multicast Service (eMBMS) ファシリティ	212000 ~ 212499
mme-misc	MME その他のファシリティ	155800 ~ 156199
mmedemux	MME Demux マネージャファシリティ	154000 ~ 154999
mmemgr	MME マネージャファシリティ	137000 ~ 137499
mmgr	マスターマネージャ (MMGR) ファシリティ	86000 ~ 86399
mobile-ip	モバイル IP (MIP) プロトコルファシリティ	26000 ~ 26999
mobile-ip-data	MIP トンネルデータファシリティ	27000 ~ 27999
mobile-ipv6	モバイル IPv6 ファシリティ	129000 ~ 129999
mpls	マルチプロトコルラベルスイッチング (MPLS) ファシリティ	163500 ~ 163999

ファシリティ	説明	イベント ID の範囲
mseg-app	Mobile Services Edge Gateway (MSEG) アプリケーション ファシリティ 本リリースではサポートされていません。	172300 ~ 172400
mseg-gtpc	MSEG GTPC アプリケーション ファシリティ 本リリースではサポートされていません。	172000 ~ 172100
mseg-gtpu	MSEG GTPU アプリケーション ファシリティ 本リリースではサポートされていません。	172200 ~ 172300
msegmgr	MSEG マネージャファシリティ 本リリースではサポートされていません。	171000 ~ 171100
mtp2	Message Transfer Part 2 (MTP2) サービスファシリティ [SS7]	116900 ~ 117000
mtp3	Message Transfer Part 3 (MTP3) サービスファシリティ [SS7]	115600 ~ 115700
multicast-proxy	マルチキャストプロキシファシリティ	94000 ~ 94100
nas	ネットワーク アクセス シグナリング (NAS) ファシリティ	153000 ~ 153100
netwstrg	ネットワーク ストレージファシリティ	78000 ~ 78100
npuctrl	ネットワーク処理ユニット (NPU) 制御ファシリティ	16000 ~ 16100
npudrv	NPU ドライバファシリティ	191000 ~ 191100
npumgr	NPU マネージャ (NPUMGR) ファシリティ	17000 ~ 17100
npumgr-acl	NPUMGR ACL ファシリティ	169000 ~ 169100
npumgr-drv	NPUMGR ドライバファシリティ	185000 ~ 185100
npumgr-flow	NPUMGR フローファシリティ	167000 ~ 167100
npumgr-fwd	NPUMGR 転送ファシリティ	168000 ~ 168100
npumgr-init	NPUMGR 初期化ファシリティ	164000 ~ 164100
npumgr-lc	NPUMGR LC ファシリティ	180000 ~ 180100
npumgr-port	NPUMGR ポートファシリティ	166000 ~ 166100
npumgr-recovery	NPUMGR リカバリファシリティ	165000 ~ 165100
npumgr-vpn	NPUMGR VPN ファシリティ	181000 ~ 181100
npusim	NPUSIM ファシリティ	176000 ~ 176100
ntfy-intf	イベント通知インターフェイス ファシリティ	170000 ~ 170100
orbs	orbs : オブジェクト リクエスト ブローカ (ORB) システムファシリティ	15000 ~ 150100

ファシリティ	説明	イベント ID の範
ospf	Open Shortest Path First (OSPF) プロトコルファシリティ	38000 ~ 38999
ospfv3	OSPFv3 プロトコルファシリティ [IPv6]	150000 ~ 150999
p2p	ピアツーピア (P2P) ファシリティ	146000 ~ 146999
pccmgr	Policy Charging and Control (PCC) マネージャファシリティ	159000 ~ 159499
pdg	Packet Data Gateway (PDG) ファシリティ	152010 ~ 152999
pdgdmgr	PDG TCP Demux Manager (pdgdmgr) ファシリティ (顧客固有のファシリティ)	162400 ~ 162999
pdif	Packet Data Interworking Function (PDIF) ファシリティ	120000 ~ 120999
pgw	Packet Data Network Gateway (PGW) ファシリティ	139000 ~ 139999
pmm-app	パケット モビリティ マネージメント (PMM) アプリケーション ファシリティ [SGSN]	89200 ~ 89499
ppp	Point-To-Point Protocol (PPP) ファシリティ	25000 ~ 25999
pppoe	Point-to-Point Protocol over Ethernet (PPPoE) ファシリティ	183000 ~ 183999
ptt	PTT ファシリティ	76000 ~ 76999
push	PUSH (VPNMgr CDR Push) ファシリティ	133000 ~ 133999
radius-acct	RADIUS アカウンティング プロトコル ファシリティ	24000 ~ 24999
radius-auth	RADIUS 認証プロトコルファシリティ	23000 ~ 23999
radius-coa	RADIUS 許可の変更 (CoA) と接続解除ファシリティ	70000 ~ 70999
ranap	Radio Access Network Application Part (RANAP) ファシリティ	87700 ~ 87899
rct	リカバリ制御タスク (RCT) ファシリティ	13000 ~ 13999
rdt	リダイレクタタスク (RDT) ファシリティ	67000 ~ 67999
resmgr	Resource Manager (RM) ファシリティ	14000 ~ 14999
rf-diameter	Rf Diameter メッセージファシリティ	92860 ~ 92869
rip	Routing Information Protocol (RIP) ファシリティ	35000 ~ 35999
rohc	Robust Header Compression (ROHC) プロトコルファシリティ	103000 ~ 103999
rsvp	RSVP プロトコルファシリティ	93000 ~ 93999
rua	RANAP User Adaptation (RUA) プロトコルファシリティ	152000 ~ 152000
slap	S1 アプリケーションプロトコル (S1AP) ファシリティ	155200 ~ 155799
saegw	System Architecture Evolution ゲートウェイファシリティ	191000 ~ 191999
sccp	Signalling Connection Control Part (SCCP) プロトコルファシリティ [SS7]	86700 ~ 86899

ファシリティ	説明	イベント ID
sct	共有設定タスク (SCT) ファシリティ	32000 ~ 32099
sctp	Stream Control Transmission Protocol (SCTP) プロトコルファシリティ	87300 ~ 87499
sess-gr	SESS-GR ファシリティ	77600 ~ 77999
sessctrl	セッションコントローラ ファシリティ	8000 ~ 8999
sessmgr	セッションマネージャ ファシリティ	10000 ~ 12999
sesstrc	セッショントレース ファシリティ	155000 ~ 155999
sft	スイッチファブリックタスク (SFT) ファシリティ	58000 ~ 58999
sgs	SG インターフェイスプロトコルファシリティ [MME]	173000 ~ 173999
sgsn-app	SGSN アプリケーションインターフェイス ファシリティ	115900 ~ 115999
sgsn-failures	SGSN コール失敗ファシリティ	89100 ~ 89199
sgsn-gtpc	SGSN GTP-C プロトコルファシリティ	116000 ~ 116999
sgsn-gtpu	SGSN GTP-U プロトコルファシリティ	86900 ~ 87099
sgsn-mbms-bearer	SGSN MBMS ベアラアプリケーション (SMGR) ファシリティ	116600 ~ 116699
sgsn-misc	SGSN その他のファシリティ	88800 ~ 89099
sgsn-system	SGSN システム コンポーネント ファシリティ	86400 ~ 86499
sgsn-test	SGSN テストファシリティ	88700 ~ 88799
sgsn2	SGSN2 ファシリティ	114000 ~ 114999
sgtpcmgr	SGSN GTP-C (SGTPC) マネージャファシリティ	117000 ~ 117999
sgw	サービングゲートウェイ (SGW) ファシリティ	140000 ~ 140999
sh-diameter	Sh Diameter メッセージファシリティ	92850 ~ 92899
sipcdprt	SIPCDPRT ファシリティ	95000 ~ 95999
sitmain	システム開始タスク (SIT) メインファシリティ	4000 ~ 4999
sm-app	ショートメッセージサービス (SMS) ファシリティ	88300 ~ 88499
sms	SMS サービスファシリティ	116800 ~ 116899
sndcp	Sub Network Dependent Convergence Protocol (SNDCP) ファシリティ	115800 ~ 115899
snmp	Simple Network Management Protocol (SNMP) ファシリティ	22000 ~ 22999
sprmgr	Subscriber Policy Register (SPR) マネージャファシリティ	159500 ~ 159599
srdb	スタティック評価データベースファシリティ	102000 ~ 102999
srp	サービス冗長性プロトコル (SRP) ファシリティ	84000 ~ 84999

ファシリティ	説明	イベント ID の範
sscfnni	SSCFNNI プロトコルファシリティ [ATM]	115500 ~ 115599
sscop	SSCOP Protocol ファシリティ [ATM]	115400 ~ 115499
ssh-ipsec	SSH IP セキュリティファシリティ	56999 ~ 56999
ssl	SSL ファシリティ (顧客固有のファシリティ)	156200 ~ 157199
stat	統計情報ファシリティ	31000 ~ 31999
system	システムファシリティ	1000 ~ 1999
tacacs+	TACACS+ プロトコルファシリティ	37000 ~ 37999
taclcp	TACLCP ファシリティ	44000 ~ 44999
tcap	Transaction Capabilities Application Part (TCAP) プロトコルのロギングファシリティ [SS7]	86500 ~ 86699
testctrl	テスト コントローラ ファシリティ	174000 ~ 17499
testmgr	テスト マネージャ ファシリティ	175000 ~ 17599
threshold	しきい値ファシリティ	61000 ~ 61999
ttg	Tunnel Termination Gateway (TTG) ファシリティ	130000 ~ 13099
tucl	TCP/UDP コンバージェンスレイヤ (TUCL) ファシリティ [SS7]	88500 ~ 88699
udr	ユーザーデータレコード (UDR) ファシリティ	79000 ~ 79999
user-data	ユーザーデータファシリティ	51000 ~ 51999
user-l3tunnel	ユーザー L3 トンネルファシリティ	75000 ~ 75999
usertcp-stack	ユーザー TCP スタックファシリティ	173300 ~ 17349
vim	ボイス インスタント メッセージ (VIM) ファシリティ	60000、60999
vinfo	VINFO ファシリティ	82000、82999
vmgctrl	仮想メディアゲートウェイ (VMG) コントローラファシリティ	41000、41999
vmgctxmgr	VMG コンテキスト マネージャ ファシリティ	43000、43999
vpn	バーチャル プライベート ネットワーク (VPN) ファシリティ	5000 ~ 5999
wimax-data	WiMAX データファシリティ	104900 ~ 10499
wimax-r6	WiMAX R6 プロトコル (シグナリング) ファシリティ	104000 ~ 10489

## イベントのシビラティ (重大度)

システムにより、ロギングが有効になっている場合に表示される情報のレベルを柔軟に設定できます。次のレベルがサポートされます。

- **クリティカル**：システムまたはシステムのコンポーネントの機能を停止させる重大なエラーが発生したことをログに記録します。これが最高のシビラティ（重大度）レベルです。
- **エラー**：システムまたはシステムのコンポーネントの動作を低下させるエラーが発生したことをログに記録します。このレベルは、シビラティ（重大度）レベルが高いイベントもログに記録します。
- **警告**：潜在的な問題を示す可能性があるイベントをログに記録します。このレベルは、シビラティ（重大度）レベルが高いイベントもログに記録します。
- **異常**：きわめて異常であり、調査する必要がある可能性のあるイベントをログに記録します。このレベルは、シビラティ（重大度）レベルが高いイベントもログに記録します。
- **情報**：シビラティ（重大度）がより高い情報イベントおよびイベントをログに記録します。
- **トレース**：トレースに役立つイベントやより高いシビラティ（重大度）を持つイベントをログに記録します。
- **デバッグ**：シビラティ（重大度）に関係なく、すべてのイベントをログに記録します。

上記のレベルはそれぞれ、イベント ID の「シビラティ（重大度）」レベルに対応します。したがって、「シビラティ（重大度）」レベルがロギングレベルと等しいイベント ID のみが表示されます。

## ログ出力のイベント ID 情報の概要

この項では、ロギングが有効になっておりときに表示されるイベント情報について説明します。

次に、ログに記録されたイベントの出力例を示します。

```
2011-Dec-11+5:18:41.993 [cli 30005 info] [8/0/609 cli:8000609 _commands_cli.c:1290]
[software internal system] CLI session ended for Security Administrator admin on device
/dev/pts/2
```

次の表で、出力例に含まれている要素について説明します。

表 15: イベント要素の説明

要素	説明
2011-Dec-11+5:18:41.993	イベントが生成された日時を示す日付/タイムスタンプ

要素	説明
[cli 30005 info]	以下を含むイベントに関する情報 <ul style="list-style-type: none"><li>• イベントが属している機能</li><li>• イベント ID</li><li>• イベントのシビラティ（重大度）</li></ul> この例では、イベントはCLIファシリティに属しており、IDは3005、シビラティ（重大度）は「info」です。
[8/0/609 cli:8000609 _commands_cli.c:1290]	特定のCLIインスタンスに関する情報。
[software internal system]	イベントの発生原因がシステムの動作であることを示しています。
CLI session ended for Security Administrator admin on device /dev/pts/2	イベントの詳細。イベントの詳細には、イベントの発生に固有の変数が含まれている場合と含まれない場合があります。





## 第 17 章

# トラブルシューティング

この章では、システムの動作中に発生する可能性のある問題をトラブルシューティングするために、システムのコマンドラインインターフェイス (CLI) を使用する方法について説明します。

- ネットワーク接続の確認 (249 ページ)
- システム診断ユーティリティの使用 (252 ページ)
- SSD の生成 (256 ページ)
- サポートデータコレクターの設定と使用 (257 ページ)
- ハイパーバイザによる強制再起動の開始 (257 ページ)

## ネットワーク接続の確認

システムでサポートされているコマンドは複数あり、ネットワーク接続の確認やトラブルシューティングを行うことができます。ネットワーク接続は、システムインターフェイスとポートが設定され、バインドされている場合にのみテストできることに注意してください。

このセクションで指定するコマンドは、コンテキストごとに発行する必要があります。コンテキストは、他のコンテキストとは独立して動作するバーチャルプライベート ネットワーク (VPN) のように機能します。あるコンテキストで設定されたポート、インターフェイス、およびルートは、追加の設定なしで別のコンテキストからテストすることはできません。

コンテキストを切り替えるには、Exec モードのルートプロンプトで次のコマンドを入力します。

```
[local]host_name# context context_name
```

*context\_name* は、切り替え先のコンテキストの名前です。次のプロンプトが表示されます。

```
[context_name]host_name#
```

## ping コマンド または ping6 コマンドの使用

**ping** または **ping6** コマンドは、応答間でデータパケットを受け渡し、測定することによって、ネットワーク内のリモートノードとシステムが通信できることを確認します。このコマンド

は、ネットワークルーティングを確認したり、リモートノードが IP レイヤで応答できるかどうかを確認するのに役立ちます。

VPC-DI の展開では、Exec モードコマンドの **system ping** を使用して、VPC-DI VM 間の内部ネットワーク (DI ネットワーク) 接続を確認します。詳細については、『*Command Line Interface Reference*』を参照してください。

## 構文

**ping** のコマンドシンタックスは、次のとおりです。

```
ping host_ipv4_address [ count num_packets ] [ flood ] [ pattern packet_pattern ] [ size octet_count ] [ src { src_host_name | src_host_ipv4_address } ] [ vrf vrf_nam ]
```

```
ping6 host_ipv6_address [ count num_packets ] [ flood ] [ pattern packet_pattern ] [ size octet_count ] [ src { src_host_name | src_host_ipv6_address } ] [ vrf vrf_nam ]
```

上記のコマンドの詳細については、『*Command Line Interface Reference*』の「*Exec Mode Commands*」の章を参照してください。

次に、成功した **ping** (IPv4) の応答の例を示します。

```
PING 209.165.200.227 (209.165.200.227): 56 data bytes
64 bytes from 209.165.200.227: icmp_seq=0 ttl=255 time=0.4 ms
64 bytes from 209.165.200.227: icmp_seq=1 ttl=255 time=0.2 ms
64 bytes from 209.165.200.227: icmp_seq=2 ttl=255 time=0.2 ms
64 bytes from 209.165.200.227: icmp_seq=3 ttl=255 time=0.2 ms
64 bytes from 209.165.200.227: icmp_seq=4 ttl=255 time=0.2 ms
--- 209.165.200.227 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 0.2/0.2/0.4 ms
```

## トラブルシューティング

ターゲットから応答を受信しない場合は、次のトラブルシューティング手順に従ってください。

- 正しい IP アドレスが入力されていることを確認します。
- 同じネットワーク上の別のデバイスに **ping** を試行します。 **ping** が成功した場合は、システム設定が正しいと思われます。 **ping** を実行しようとしているデバイスの電源が入っており、正常に機能していることを確認します。
- ポートが動作していることを確認します。
- コンテキスト内のポートとインターフェイスの設定が正しいことを確認します。
- 設定が正しく、 **ping** を試行しているデバイスにアクセスできる場合は、そのデバイスからシステムに **ping** を実行します。
- まだ応答がない場合は、パケットがネットワークデバイスによって破棄されている可能性があります。この章で説明されている **traceroute** コマンドまたは **traceroute6** コマンドおよび **show ip static-route** コマンドを使用して、この問題のトラブルシューティングを行います。

## traceroute または traceroute6 コマンドの使用

**traceroute** コマンドまたは **traceroute6** コマンドは、指定されたホストに送信されるルートデータに関する情報を収集します。これは、ネットワーク上の重大なパケット遅延またはパケット損失の原因を特定するために使用できる、便利なトラブルシューティング コマンドです。また、このコマンドは、ネットワークを介したデータのルーティングのボトルネックを識別するためにも使用できます。

### traceroute : IPv4

次に、**traceroute** コマンドのシンタックスを示します。

```
traceroute { host_name | host_ipv4_address } [ count packets ] [ df ] [ maxttl max_ttl ] [ minttl min_ttl ] [ port port_number ] [ size octet_count ] [ src { src_host_name | src_host_ipv4_address } ] [ timeout seconds ] [ vrf vrf_name ]
```

上記のコマンドの詳細については、『*Command Line Interface Reference*』の「*Exec Mode Commands*」の章を参照してください。

次に、出力例を示します。

```
traceroute to 209.165.200.227 (209.165.200.227), 30 hops max, 40 byte packets
 1 209.165.200.227 (209.165.200.227) 0.446 ms 0.235 ms 0.178 ms
```

### traceroute6 : IPv6

次に、**traceroute6** コマンドのシンタックスを示します。

```
traceroute6 { host_name | host_ipv6_address } [ count packets ] [ maxttl max_ttl ] [ port port_number ] [ size octet_count ] [ src { src_host_name | src_host_ipv6_address } ] [ timeout seconds ] [ vrf vrf_name ]
```

上記のコマンドの詳細については、『*Command Line Interface Reference*』の「*Exec Mode Commands*」の章を参照してください。

次に、出力例を示します。

```
traceroute6 to 2001:4A2B::1f3F (2001:4A2B::1f3F), 30 hops max, 40 byte packets
 1 2001:4A2B::1f3F (2001:4A2B::1f3F) 0.446 ms 0.235 ms 0.178 ms
```

## IP ルートの表示

このシステムには、特定のノードへのルート情報またはコンテキスト全体を表示するメカニズムが備わっています。この情報を使用して、ネットワーク接続を確認し、ネットワーク接続の効率を高めることができます。コマンドの構文は、次のとおりです。

```
show ip route [ route_ip_address ]
show ipv6 route [ route_ipv6_address ]
```

上記のコマンドの詳細については、『*Command Line Interface Reference*』の「*Exec Mode show Commands*」の章を参照してください。

キーワードを指定しなかった場合は、コンテキストのルーティングテーブル内のすべての IP ルートが表示されます。

次に、コンテキスト IPv4 ルーティングテーブルが示されているこのコマンドの出力例を示します。

```

**" indicates the Best or Used route.
  Destination      Nexthop      Protocol    Prec    Cost    Interface
*0.0.0.0/0         10.0.4.1     static      0       0       SPI01
*10.0.4.0/24       0.0.0.0     kernel      0       0       SPI01
*10.0.4.0/32       0.0.0.0     kernel      0       0       SPI01
*10.0.4.3/32       0.0.0.0     kernel      0       0       SPI01
*10.0.4.255/32    0.0.0.0     kernel      0       0       SPI01
    
```

## アドレス解決プロトコルテーブルの表示

システムは、特定のノードまたはコンテキスト全体に対して、Address Resolution Protocol (ARP) のテーブル情報を表示するメカニズムを提供します。この情報は、システムが ARP パケットを送信したときに、他のネットワークノードから有効な応答を受信したことを確認するために使用できます。

```
[local]host_name# show ip arp [ arp_ip_address ]
```

*arp\_ip\_address* は、ARP 情報を表示する特定のネットワークノードを指定します。このアドレスは、IPv4 ドット付き 10 進表記または IPv6 コロン区切り 16 進表記で入力できます。このキーワードが指定されていない場合は、コンテキストの ARP テーブル内のすべてのエントリが表示されます。



**重要** VPN マネージャを再起動すると、カーネルからすべてのインターフェイスが削除されます。これにより、すべての ARP エントリが削除されます。ただし、NPU では、トラフィックが中断されないように、すべての ARP エントリが引き続き保持されます。ユーザーの観点から、このコマンドは NPU ではなくカーネルから情報を収集するため、**show ip arp** が破損しています。

次に、コンテキストの ARP テーブルを表示するこのコマンドの出力例を示します。

```

Flags codes:
C - Completed, M - Permanent, P - Published, ! - Not answered
T - has requested trailers
Address      Link Type    Link Address      Flags    Mask Interface
10.0.4.240   ether        00:05:47:02:20:20 C        MIO1
10.0.4.7     ether        00:05:47:02:03:36 C        MIO1
10.0.4.1     ether        00:01:30:F2:7F:00 C        MIO1
    
```

## システム診断ユーティリティの使用

システムには、設定のトラブルシューティングや確認の際に役立つプロトコルモニターとテストユーティリティが備わっています。これらのユーティリティによって生成される情報は、ソフトウェアやネットワーク設定の問題の根本原因を特定するのに便利です。

この項では、これらのユーティリティの使用方法について説明します。



**重要** この章で説明する診断ユーティリティは、オペレータ以上の権限を持つ管理者のみが実行できます。

## モニターユーティリティの使用

システムにはトラブルシューティングを目的としたプロトコル モニタリング ユーティリティが用意されています。このツールは、特定のサブスクライバセッションか、または処理中のすべてのセッションのプロトコル情報を表示します。



**注意** モニターツールによって、セッションの処理遅延やデータ損失が発生する場合があります。したがって、トラブルシューティングを行う場合にのみ使用してください。

## プロトコルモニターの使用

プロトコルモニターには、現在処理中のすべてのセッションの情報が表示されます。モニター対象のプロトコルの数と進行中のセッション数に応じて、大量のデータが生成されます。生成されたすべての情報をキャプチャするには、端末クライアントでロギングを有効にすることを強くお勧めします。

**monitor protocol** コマンドおよび **monitor subscriber** コマンドの PCAP 機能を有効にするには、[パケットキャプチャ \(PCAP\) トレース \(259 ページ\)](#) も参照してください。

プロトコル モニタリング ツールを起動して設定するには、次の手順に従います。

**ステップ 1** **monitor protocol** コマンドを入力して、Exec モードでプロトコルモニターを起動します。

```
[local]host_name# monitor protocol
```

現在使用可能なすべてのプロトコル（それぞれに割り当てられた番号を持つ）が一覧表示された出力が表示されます。

**ステップ 2** *Select*: プロンプトで関連付けられた番号を入力して、モニターするプロトコルを選択します。選択したプロトコルの横に右矢印 (>) が表示されます。

**ステップ 3** 必要に応じてステップ 2 を繰り返して、複数のプロトコルを選択します。

**ステップ 4** **B** を押して、プロトコルモニターを開始します。

```
WARNING!!! You have selected options that can DISRUPT USER SERVICE
Existing CALLS MAY BE DROPPED and/or new CALLS MAY FAIL!!!
(Under heavy call load, some debugging output may not be displayed)
Proceed? - Select (Y)es or (N)o
```

**ステップ 5** **Y** を入力してモニターを続行するか、**N** を入力して前のメニューに戻ります。

```
C - Control Events      (ON )
D - Data Events        (ON )
```

```

E - EventID Info           (ON )
H - Display ethernet      (ON )
I - Inbound Events        (ON )
O - Outbound Events       (ON )
S - Sender Info           (OFF)
T - Timestamps            (ON )
X - PDU Hexdump           (OFF)
A - PDU Hex/Ascii         (OFF)
+/- Verbosity Level       (  1)
L - Limit Context         (OFF)
M - Match Newcalls        (ON )
R - RADIUS Dict           (no-override)
G - GTPP Dict             (no-override)
Y - Multi-Call Trace      ((OFF))
(Q)uit,      <ESC> Prev Menu,      <SPACE> Pause,      <ENTER> Re-Display Options

```

**ステップ6** モニターによって表示される情報の量を設定します。オプションを有効化または無効化するには、そのオプションに関連付けられている文字（C、D、E など）を入力します。冗長性を向上または低下させるには、プラス（+）またはマイナス（-）キーを使用します。

各オプションの右側には、[ON (enabled)] または [OFF (disabled)] の現在の状態が表示されます。

**ステップ7** [Enter] キーを押して画面を更新し、モニタリングを開始します。

---

モニターは、無効になるまでアクティブのままになります。プロトコルモニターを終了してプロンプトに戻るには、**q** を押します。

## 特定サブスクリイバのプロトコルモニターの使用

プロトコルモニターは、現在処理中の特定のサブスクリイバセッションの情報を表示するために使用できます。モニター対象のプロトコルの数と進行中のセッション数に応じて、大量のデータが生成されます。生成されたすべての情報をキャプチャするには、端末クライアントでロギングを有効にすることを強くお勧めします。

特定のサブスクリイバセッションのプロトコルモニタリングツールを起動して設定するには、この項の手順に従います。

---

**ステップ1** Exec モードからセッション固有のプロトコルモニターを起動するには、**monitor subscriber** コマンドを入力します。

```

[local]host_name# monitor subscriber { callid | imei | imsi | ipaddr | ipv6addr |
msid | msisdn | next-call | pcf | peer-fa | peer-lac | sgsn-address | type |
username }

```

**ステップ2** 適切なキーワードを入力して、モニターが使用するメソッドを指定します。

**ステップ3** その他のオプションを選択したり、選択したキーワードに適切な情報を入力したりします。

モニターの起動時に、指定された基準に一致するセッションが処理されなかった場合は、使用可能なモニタリングオプションの画面が表示されます。

**ステップ4** モニターによって表示される情報の量を設定します。オプションを有効または無効にするには、そのオプションに関連付けられている文字または2桁の数字（C、D、E、11、12 など）を入力します。冗長性を向上または低下させるには、プラス（+）またはマイナス（-）キーを使用します。

各オプションの右側には、[ON (enabled)] または [OFF (disabled)] の現在の状態が表示されます。

マルチコールドレースを実行するためのオプション **Y** は、GGSN での使用に対してのみサポートされています。

**ステップ 5** 必要に応じて **ステップ 6** を繰り返して、複数の Protokol を有効または無効にします。

**ステップ 6** **Enter** を押して画面を更新し、モニタリングを開始します

次に、*user2@aaa* という名前のサブスクリバに対するモニターの出力例の一部を示します。デフォルトの Protokol がモニターされました。

```
-----
Incoming Call:
-----
MSID: 0000012345 Callid: 002dc6c2
Username: user2@aaa SessionType: unknown
Status: Active Service Name: xxx1
Src Context: source Dest Context:
-----

<<<<OUTBOUND 10:02:35:415 Eventid:25001(0)
PPP Tx PDU (9)
PAP 9: Auth-Ack(1), Msg=

<<<<OUTBOUND 10:02:35:416 Eventid:25001(0)
PPP Tx PDU (14)
IPCP 14: Conf-Req(1), IP-Addr=192.168.250.70

<<<<OUTBOUND 10:02:35:416 Eventid:25001(0)
PPP Tx PDU (27)
CCP 27: Conf-Req(1), MPPC, Stac-LZS, Deflate, MVRCA

INBOUND>>>>> 10:02:35:517 Eventid:25000(0)
PPP Rx PDU (30)
IPCP 30: Conf-Req(1), IP-Comp VJ-Comp, IP-Addr=0.0.0.0, Pri-DNS=0.0.0.0,
Sec-DNS=0.0.0.0

<<<<OUTBOUND 10:02:35:517 Eventid:25001(0)
PPP Tx PDU (26)
IPCP 26: Conf-Rej(1), IP-Comp VJ-Comp, Pri-DNS=0.0.0.0, Sec-DNS=0.0.0.0

INBOUND>>>>> 10:02:35:517 Eventid:25000(0)
PPP Rx PDU (12)
IPCP 12: Conf-Ack(1), IP-Addr=192.168.250.70

INBOUND>>>>> 10:02:35:518 Eventid:25000(0)
PPP Rx PDU (31)
LCP 31: Prot-Rej(1), Rejected-Protocol=CCP (0x80fd)

INBOUND>>>>> 10:02:35:518 Eventid:25000(0)
PPP Rx PDU (12)
IPCP 12: Conf-Req(2), IP-Addr=0.0.0.0

<<<<OUTBOUND 10:02:35:518 Eventid:25001(0)
PPP Tx PDU (14)
IPCP 14: Conf-Nak(2), IP-Addr=192.168.250.87

INBOUND>>>>> 10:02:35:519 Eventid:25000(0)
PPP Rx PDU (12)
IPCP 12: Conf-Req(3), IP-Addr=192.168.250.87
```

モニターは、無効になるまでアクティブのままになります。プロトコルモニターを終了してプロンプトに戻るには、**q** を押します。

## SSD の生成

SSD は、Exec モードの **show support details** コマンドが実行されたときの出力のインスタンスです。トラブルシューティングのために役立つシステム情報の包括的なリストが表示されます。ほとんどの場合、このコマンドの出力はテクニカルアシスタンスセンター (TAC) によって要求されます。

SSD 出力の .tar ファイルは、ローカルまたはリモートの場所 (URL) にリダイレクトできます。

.tar ファイルには次のものが含まれます。

- **support\_summary** : サポートの詳細情報を含む ASCII テキストファイル。
- **information.minicores.tar** : システム上で検出された minicore ファイルを含む .tar ファイル。Minicore ファイルには、一部のイベント中にキャプチャされるメモリアダンプが含まれています。これらのコアダンプは、イベントに関する特定のメモリの場所とその他の情報を提供します。この情報はテクニカルサポートチームにとって、推定原因とともにイベントが発生した場所とタイミングを特定するために役立ちます。

**show support details** コマンドには、他の方法ではユーザーがアクセスできない情報が含まれていますが、TAC による問題の迅速な解決に役立ちます。



**重要** 大規模なコンフィギュレーションファイルを持つプラットフォームでは、SSD を完了するまでに最大で30分かかる場合があります。**show support details** コマンドを実行すると、システムリソースが消費され、トラフィックのスループットが低下する可能性があります。

オペレータが **show support details** コマンドを入力したときに SSD が進行中である場合、StarOS は SSD がすでに進行中であることを示す警告メッセージで応答します。ユーザーは後で再試行する必要があります。オペレータは、一度に 1 つの SSD インスタンスだけを実行するように制限されています。

**show support details** コマンドには、特定のタイプの情報だけを報告するように SSD をターゲットにできるオプションのキーワードがあります。これらのキーワードにより、SSD の生成に必要な時間を短縮できます。

**show support details** コマンドの詳細については、『*Command Line Interface Reference*』の「*Exec Mode Show Commands (Q-S)*」の章を参照してください。



## サポートデータコレクターの設定と使用

サポートデータを収集するタスクは、record collector と呼ばれるバックグラウンド CLI タスクによって実行されます。管理者は、CLI を介して Support Data Collector (SDC) を設定し、コマンドを定期的に行います。レコードコレクタは常にバックグラウンドで実行され、収集レコードがあるかどうかを確認します。

サポートデータを収集する時間になると、スケジューラは設定された CLI コマンドのシーケンスを実行し、その結果をハードディスク上の gunzipped (gz) ファイルに保存します。このファイルは SDR (サポートデータレコード) と呼ばれ、その時点でのシステム全体の状態のスナップショットを表します。

テクニカルアシスタンスセンター (TAC) 担当者およびローカル管理者は、SDR をオンラインで、またはシステムから転送して確認することができます。また、コレクタの状態の情報を調査する場合があります。

SDC 機能の詳細については、「サポートデータコレクター」の章を参照してください。

## ハイパーバイザによる強制再起動の開始

ハイパーバイザは、仮想ウォッチドッグデバイスをサポートしています。VPCがこのウォッチドッグの機能を停止すると、ハイパーバイザによってVMが強制的に再起動されます。次の表を参照してください。

表 16: ハイパーバイザによる強制再起動の条件

条件	再起動方法	リカバリ	注意
クリティカルタスクの失敗	ハイパーバイザウォッチドッグ	ハイパーバイザによるVMの再起動	StarOS がウォッチドッグの機能を止めます。
カーネルのハング/クラッシュ	カーネルまたはハイパーバイザウォッチドッグ	ハイパーバイザによるVMの再起動	
ホストの障害	ハイパーバイザ HA (高可用性)	ハイパーバイザ管理システムが HA を呼び出し、VM を別のホストに割り当てて再起動します。	例: VMware の HA クラスタ

KVM では、`--watchdog i6300esb` コマンドライン引数を使用して仮想ウォッチドッグデバイスを指定できます。VMware には、独自のウォッチドッグメカニズムが備わっています。





## 第 18 章

# パケットキャプチャ (PCAP) トレース

- [機能情報 \(259 ページ\)](#)
- [機能説明 \(260 ページ\)](#)
- [PCAP トレースの設定 \(261 ページ\)](#)
- [PCAP トレースのモニタリングとトラブルシューティング \(269 ページ\)](#)

## 機能情報

### 要約データ

該当製品または機能エリア	<ul style="list-style-type: none"><li>• ePDG</li><li>• IPSec</li><li>• MME</li><li>• SaMOG</li></ul>
該当プラットフォーム	ASR 5500 vPC-SI vPC-DI
機能のデフォルト	無効
このリリースでの関連する変更点	N/A

関連資料	<ul style="list-style-type: none"> <li>• <i>ASR 5000 System Administration Guide</i></li> <li>• <i>ASR 5500 System Administration Guide</i></li> <li>• <i>Command Line Interface Reference Guide</i></li> <li>• <i>ePDG Administration Guide</i></li> <li>• <i>IPSec Reference Guide</i></li> <li>• <i>SaMOG Administration Guide</i></li> <li>• <i>VPC-SI System アドミニストレーション ガイド</i></li> </ul>
------	--

### マニュアルの変更履歴



**重要** リリース 21.2 よりも前に導入された機能については、詳細な改訂履歴は示していません。

改訂の詳細	リリース
このリリースでは、MMESI-AP インターフェイスに対する PCAP トレーシングのサポートが追加されました。	21.4
最初の導入。	21.2

## 機能説明

この機能により、**monitor subscriber** コマンドと **monitor protocol** コマンドの出力をパケットキャプチャ (PCAP) 機能を使用してキャプチャできます。この出力はハードディスクにテキストファイルとして保存し、後で PUSH または PULL 方式を使用し、SFTP を介して外部サーバーに転送できます。テキストファイルは、text2pcap などの外部ツールを使用して pcap ファイルに変換したり、wireshark などのパケットアナライザツールを使用して PCAP として直接インポートしたりできます。

PCAP のトレースおよび 16 進ダンプファイルの収集は、**monitor protocol** コマンドと **monitor subscriber** コマンドで有効または無効にできます。詳細については、この章の「16 進ダンプの有効化または無効化」の項を参照してください。



(注) VPC-DI の展開では、内部 DI ネットワーク上の特定のカード (VM) とカードインターフェイスでパケットのキャプチャを実行するために別の機能を使用することができます。詳細については、『*Command Line Interface Reference*』の「Exec mode command **system packet-dump command**」を参照してください。

# PCAP トレースの設定

## CDRMOD の複数のインスタンスの有効化

次の設定を使用して、CDRMOD の複数のインスタンス（パケット処理カードあたり 1 つ）を有効にします。

```
config
  cdr-multi-mode
end
```

注：

- 16 進ダンプレコード生成はシングルモードとマルチモードの両方でサポートされていますが、CDR マルチモードを有効にすることを推奨します。



---

**重要** `cdr-multi-mode` CLI コマンドを設定した後は、シャースィをリロードしてコマンドを有効にする必要があります。設定ファイルを保存してシャースィをリロードする方法については、使用している展開の『*System Administration Guide*』を参照してください。

---

- `default cdr-multi-mode` コマンドを使用してこのコマンドをデフォルト値に設定します。



---

**重要** `default cdr-multi-mode` CLI コマンドを設定した後は、シャースィをリロードしてコマンドを有効にする必要があります。設定ファイルを保存してシャースィをリロードする方法については、使用している展開の『*System Administration Guide*』を参照してください。

---

- デフォルト：シングル CDRMOD モード

## 16 進ダンプモジュールの設定

16 進ダンプファイルの処理特性を指定するには、次の設定を使用します。

```
config
  context context_name
    hexdump-module
      hexdump { purge { storage-limit megabytes | time-limit seconds }
        [ max-files max_records ] | push-interval interval | push-trigger
        space-usage-percent trigger_percent | remove-file-after-transfer |
        transfer-mode { pull [ module-only ] | push primary { encrypted-url |
        url } url [ secondary { encrypted-secondary-url | secondary-url }
        secondary_url ] [ via local-context ] [ max-files files ] [ max-tasks
```

```
max_tasks ] [ module-only ] } | use-harddisk }
end
```

注：

- キーワードをデフォルト設定するには、**default hexdump [ purge | push-interval | push-trigger [ space-usage-percent ] | remove-file-after-transfer | transfer-mode [ module-only ] | use-harddisk ]** + コマンドを使用します。
  - **purge** : 有効になっていない
  - **push-interval** : 60 秒
  - **push-trigger** : 80パーセント
  - **remove-file-after-transfer** : 無効
  - **transfer mode** : プッシュ
  - **use-harddisk** : 無効
- 設定された 16 進ダンプファイルの保存と処理を無効にするには、**no hexdump [ purge | remove-file-after-transfer | use-harddisk ]** + コマンドを使用します。
  - **purge** : ストレージ制限または時間制限に基づいて、ハードディスク上のレコードファイルの削除を無効にします。
  - **remove-file-after-transfer** : 別のサーバーにプッシュまたはプルされた後でも、ファイルのコピーを保持します。
  - **use-harddisk** : システムのハードディスク上のデータストレージを無効にします。
- ハードドライブから 16 進ダンプレコードを削除するためのパラメータを設定するには、**purge { storage-limit megabytes | time-limit seconds } [ max-files max\_records ]** キーワードを使用します。このコマンドはデフォルトでは無効になっています。
  - **storage-limit megabytes** : メガバイト単位で定義されたストレージ制限に達したときに、ハードドライブから 16 進ダンプレコードを削除するように指定します。  
bytes 10 ~ 143360 の整数である必要があります。
  - **time-limit seconds** : 秒単位で定義された時間制限に達したときに、ハードドライブから 16 進ダンプレコードを削除するように指定します。  
seconds は 600 ~ 2592000 の整数である必要があります。
  - **max-files max\_records** : 消去するファイルの最大数を指定します。0 に設定すると、制限に達するまですべてのレコードが消去されます。  
max\_records は、値 0 または 1000 ~ 10000 の整数である必要があります。
- 16 進ダンプファイルが外部ファイルサーバーにプッシュされる場合の転送間隔 (秒単位) を指定するには、**push-interval interval** キーワードを使用します。

- 間隔は 30 ~ 3600 の整数である必要があります。
- デフォルト : 60
- **push-trigger space-usage-percent** *trigger\_percent* を使用して、自動プッシュがトリガーされファイルが外部サーバーに転送されるディスク容量の使用率 (パーセンテージ) のしきい値を指定します。
  - *trigger\_percent* は 10 ~ 80 の整数である必要があります。
  - デフォルトは 80 です。
- システムが外部ファイルサーバーに転送された後に 16 進ダンプファイルを削除する必要があることを指定するには、**remove-file-after-transfer** キーワードを使用します。  
デフォルトで、ディセーブルになっています。



**重要** このキーワードは、16 進ダンプレコードに対して有効にする必要があります。

- **transfer-mode { pull [ module-only ] | push primary { encrypted-url | url } Url [ secondary { encrypted-secondary-url | secondary-url } secondary\_url ] [ via local-context ] [ max-files files ] [ max-tasks max\_tasks ] [ module-only ] }** キーワードを使用して、外部ファイルサーバーに 16 進ダンプファイルを転送するときに使用する転送モードを指定します。
  - **pull** : 宛先サーバー (外部ストレージ) が 16 進ダンプファイルをプルするように指定します。
  - **push** : システムが 16 進ダンプファイルを宛先サーバーにプッシュするように指定します。これは、デフォルトのモードです。
  - **primary encrypted-url url** : システムがファイルを暗号化形式でプッシュするプライマリ URL の場所を指定します。  
*url* は、1 ~ 8192 文字の英数字の文字列である必要があります。
  - **primary url url** : システムが 16 進ダンプファイルをプッシュするプライマリ URL の場所を指定します。  
*url* は次のフォーマットの 1 ~ 1024 文字の英数字文字列になっている必要があります。  
`//user:password@host:[port]/direct`。
  - **secondary encrypted-secondary-url secondary\_url** : システムがファイルを暗号化形式でプッシュするセカンダリ URL の場所を指定します。  
*secondary\_url* は、1 ~ 8192 文字の英数字の文字列である必要があります。
  - **secondary secondary-url secondary\_url** : システムが 16 進ダンプファイルをプッシュするセカンダリ URL の場所を指定します。

`secondary_url` は、次の形式の 1 ~ 1024 文字の英数字の文字列である必要があります。`/user:password@host:[port]/direct`。

- **via local-context** : ローカルコンテキスト、およびその後の SPIO 管理ポートを使用して、16 進ダンプファイルをプルまたはプッシュするように指定します。
- **max-files files** : プッシュごとに転送できるファイルの最大数を指定します。  
ファイルは 4 ~ 4000 の整数である必要があります。
- **max-tasks max\_tasks** : プッシュごとの最大ファイル数を指定します。  
`max_tasks` は 4 ~ 8 の整数である必要があります。
- **module-only** : 16 進ダンプレコードの転送を、設定が最初に作成されたモジュールのタイプにのみ適用するように指定します。このオプションが有効になっていない場合は、すべてのレコードタイプに対して転送が行われます。
- 16 進ダンプレコードを保存するために SMC 上のハードディスクドライブを使用するように指定するには、**use-harddisk** キーワードを使用します。  
デフォルトで、ディセーブルになっています。




---

**重要** このキーワードは、16 進ダンプレコードに対して有効にする必要があります。

---

## 16 進ダンプファイルパラメータの設定

16 進ダンプファイルの形式を指定するには、次の設定を使用します。

```
config
  context context_name
  hexdump-module
    file [ compression { gzip | none } | current-prefix prefix |
delete-timeout seconds | directory directory_name | exclude-checksum-record
| field-separator { hyphen | omit | underscore } | headers | name
file_name | reset-indicator | rotation { num-records number | tariff-time
minute minutes hour hours | time seconds | volume bytes } |
sequence-number { length length | omit | padded | padded-six-length |
unpadded } | storage-limit limit | time-stamp { expanded-format |
rotated-format | unix-format } | trailing-text string |
trap-on-file-delete | xor-final-record ] +
  end
```

注 :

- 指定されたキーワードのデフォルト設定を設定するには、**default file [ compression | current-prefix | delete-timeout | directory | field-separator | headers | name | reset-indicator |**



**rotation { num-records | tariff-time | time | volume } | sequence-number | storage-limit | time-stamp | trailing-text | trap-on-file-delete ]** + コマンドを使用します。

- 16 進ダンプファイルの圧縮を指定するには、**compression { gzip | none }** キーワードを使用します。
  - **gzip** : 約 10:1 の比率で 16 進ダンプファイルの GNU zip 圧縮を有効にします。
  - **none** : Gzip 圧縮を無効にします。
- レコードを保存するために現在使用されている 16 進ダンプファイルの先頭に追加する文字列を指定するには、**current-prefix prefix** キーワードを使用します。
  - **prefix** は、1 ~ 31 文字の英数字の文字列である必要があります。
  - デフォルト : `curr`
- 16 進ダンプファイルが削除されるまでの期間を秒単位で指定するには、**delete-timeout seconds** キーワードを使用します。デフォルトでは、ファイルは削除されません。
  - **seconds** は 3600 ~ 31536000 の整数である必要があります。
  - デフォルト : `[Disabled]`
- 16 進ダンプファイルを保存するデフォルトのディレクトリのサブディレクトリを指定するには、**directory directory\_name** キーワードを使用します。
  - **directory\_name** は、0 ~ 191 文字の英数字の文字列である必要があります。
  - デフォルト : `/records/hexdump`
- **exclude-checksum-record** キーワードを使用して、`#CHECKSUM` に続いて先行するすべてのレコードの 32 ビット Cyclic Redundancy Check (CRC; 巡回冗長検査) を含む最終レコードを、16 進ダンプファイルから除外します。

デフォルト : 無効 (チェックサムレコードは 16 進ダンプファイルのヘッダーに含まれる)
- 16 進ダンプファイル名の 2 つのフィールド間における区切り文字のタイプを指定するには、**field-separator { hyphen | omit | underscore }** を使用します。
  - **hyphen** : フィールドの区切り文字を、2 つのフィールド間の「-」 (ハイフン) 記号として指定します。
  - **omit** : 2 つのフィールドの間でフィールドの区切り文字を省略します。
  - **underscore** : フィールドの区切り文字を、2 つのフィールド間の「\_」 (下線) 記号として指定します。
- レコードレイアウトを要約しているファイルヘッダーを含めるには、**headers** キーワードを使用します。
- 16 進ダンプファイルのベースファイル名として使用する文字列を指定するには、**name file\_name** を使用します。

*file\_name* は、1 ～ 31 文字の英数字の文字列である必要があります。

- 16 進ダンプファイル名内にリセット インジケータ カウンタ (0 ～ 225 の値) を含めることを指定するには、**reset-indicator** を使用します。  
カウンタは、次のいずれかの条件が発生するたびに増分されます。
  - シャーシ間セッションリカバリ (ICSR) に従って、ピアシャーシが引き継がれた場合。
  - シーケンス番号 (**sequence-number** キーワードを参照) がゼロにロールオーバーされた場合。
- 16 進ダンプファイルを閉じて新しいファイルを作成するタイミングを指定するには、**rotation { num-records number | tariff-time minute minutes hour hours | time seconds | volume bytes }** キーワードを使用します。
  - **num-records number** : 16 進ダンプファイルに追加する必要があるレコードの最大数を指定します。ファイル内のレコード数がこの値に達すると、ファイルが完成します。  
*number* は 100 ～ 10240 の整数である必要があります。デフォルト : 1024
  - **tariff-time minute minutes hour hours** : 現在の 16 進ダンプファイルを閉じて、タリフ時間 (時分単位) に基づいて新しいファイルを作成します。  
*minutes* は 0 ～ 59 の整数である必要があります。  
*hours* は 0 ～ 23 の整数である必要があります。
  - **time seconds** : 現在の 16 進ダンプファイルを閉じて新しいファイルを作成するまでに待機する時間 (秒単位) を指定します。  
*seconds* は 30 ～ 86400 の整数である必要があります。デフォルト : 3600



**重要** ローテーション時間は 30 秒に設定することを推奨します。

- **volume bytes** : 16 進ダンプファイルを閉じて、新しいファイルを作成するまでのファイルの最大サイズを指定します (バイト単位)。  
*bytes* は 51200 ～ 62914560 の整数である必要があります。compression キーワードが gzip に設定されている場合、設定が大きいほど圧縮率が向上する可能性があることに注意してください。デフォルト : 102400
- 指定した形式でファイル名内からシーケンス番号を除外するか、またはファイル名内を含めるには、**sequence-number { length length | omit | padded | padded-six-length | unpadded }** キーワードを使用します。
  - **length length** : 指定した長さのシーケンス番号が含まれます。  
*length* はファイル名の前に 0 を付けたファイルシーケンス番号の長さであり、また、1 ～ 9 の整数である必要があります。

- **omit** : ファイル名からシーケンス番号を除外します。
  - **padded** : ファイル名の前に 0 を追加してパディングしたシーケンス番号が含まれます。これがデフォルトの設定です。
  - **padded-six-length** : ファイル名の前に 6 つゼロを追加してパディングしたシーケンス番号を含めます。
  - **unpadded** : ファイル名にパディングのないシーケンス番号を含めます。
- ストレージ制限を設定するには、**storage-limit limit** キーワードを使用します。指定した領域の量 (バイト単位) に達すると、ファイルは削除されます。
- limit* は 10485760 ~ 268435456 の整数である必要があります。
- ファイル名に含めるファイル作成タイムスタンプの形式を指定するには、**time-stamp { expanded-format | rotated-format | unix-format }** キーワードを使用します。
- **expanded-format** : UTC (協定世界時) の MMDDYYYYHHMMSS 形式を指定します。
  - **rotated-format** : タイムスタンプの形式を YYYYMMDDHHMMSS 形式に指定します。
  - **unix-format** : UNIX 形式の x.y を指定します。ここで、x は 1970 年 1 月 1 日からの秒数で、y は経過した現在の秒数の小数部分です。
- ファイル名に任意のテキスト文字列を 1 ~ 30 文字の英数字の文字列として含めるには、**trailing-text string** キーワードを指定します。
- string* は、1 ~ 30 文字の英数字の文字列である必要があります。
- 16 進ダンプファイルが領域不足のために削除されたときに SNMP 通知 (トラップ) を送信するようにシステムに指示するには、**trap-on-file-delete** キーワードを使用します。
- デフォルト : [Disabled]
- **exclude-checksum-record** がデフォルト設定のままになっている場合に、排他的論理和 (XOR) チェックサムを (CRC チェックサムの代わりに) 16 進ダンプファイルのヘッダーに挿入するには、**xor-final-record** キーワードを使用します。
- デフォルト : [Disabled]
- この + 記号は、1 つのコマンド内で前のキーワードの 1 つ以上を入力できることを示します。

## 16 進ダンプの有効化または無効化

16 進ダンプキャプチャは、Exec モードの **monitor subscriber** コマンドおよび **monitor protocol** コマンドでプロトコルに対して有効にできます。PCAP トレースのサブスクライバ情報は、**monitor subscriber** コマンドのフィルタを使用して指定できます。特定の製品でサポートされているプロトコルおよびフィルタについては、それぞれの製品管理およびリファレンスガイドを参照してください。

**monitor subscriber** コマンドまたは **monitor protocol** コマンドが実行されている場合は、**U** オプションまたは **V** オプションを使用して 16 進ダンプキャプチャを有効にします。

- **U - Mon Display (ON)** : 端末上のメッセージのキャプチャを表示するには、このオプションを使用します。
  - デフォルト : オン
  - このオプションをオフにすると、モニタリングは引き続きバックグラウンドで実行されます。
- **V - PCAP Hexdump (NONE)** : 16 進ダンプパケットのキャプチャをグローバルに有効または無効にするには、このオプションを使用します。
  - デフォルト : なし
  - **V - PCAP Hexdump (ON)** : 次のプロンプトで 16 進ダンプキャプチャが有効になります。
 

警告 : オン/オフを切り替えると、他の *cli* ロギング端末に影響します。すでに 16 進ダンプを使用している他のユーザーに割り込むこととなります。
  - **V - PCAP Hexdump (OFF)** : 16 進ダンプキャプチャは無効になっています (一時停止)。

## MME の PCAP トレースの有効化

この項では、MME S1-AP インターフェイスと SGSAP インターフェイスに対して PCAP トレースを有効にする方法について説明します。

- モニタープロトコル (**monpro**) で、S1-AP と SGS、または SCTP プロトコルオプションを **V - PCAP Hexdump (ON)** とともに有効にして、PCAP hexdump 内の S1-AP メッセージをキャプチャします。
- モニターサブスクライバ (**monsub**) は、S1-AP と SGS フィルタオプションで PCAP トレースをサポートしています。
- **monpro/monsub** で S1-AP または SGS フィルタオプションを選択すると、PCAP Hexdump にはダミーの SCTP ヘッダーが表示されます。SCTP ヘッダーには、次のフィールドがダミーとして設定されています。
  - 確認タグ
  - Checksum
  - チャンクフラグ
  - 送信シーケンス番号 (TSN)
  - ストリーム識別子
  - ストリームシーケンス番号

- monpro で SCTP プロトコルオプションを選択した場合、PCAP 16 進ダンプには元の SCTP ヘッダーが表示されます。

# PCAP トレースのモニタリングとトラブルシューティング

## コマンドや出力の表示

このセクションの show コマンドは、PCAP トレースのサポートに使用できます。

### show cdr statistics

この機能をサポートする **show cdr statistics** コマンドの出力では、次のフィールドを使用できます。

```
EDR-UDR file Statistics:
-----
CDRMOD Instance Id: 2
Hexdump-module Record Specific Statistics:
Hexdump-module files rotated: 0
Hexdump-module files rotated due to volume limit: 0
Hexdump-module files rotated due to time limit: 0
Hexdump-module files rotated due to tariff-time: 0
Hexdump-module files rotated due to records limit: 0
Hexdump-module file rotation failures: 0
Hexdump-module files deleted: 0
Hexdump-module records deleted: 0
Hexdump-module records received: 0
Current open Hexdump-module files: 0
Time of last Hexdump-module file deletion: 0
```

表 17: show cdr statistics コマンド出力の説明

フィールド	説明
<b>EDR-UDR ファイルの統計情報 :</b>	
CDRMOD インスタンス ID	統計情報が収集される CDRMOD のインスタンス ID を示します。
<b>Hexdump-module レコード固有の統計情報 :</b>	
Hexdump-module files rotated	16進ダンプファイルが閉じられ、新しい16進ダンプファイルが作成された合計回数。
Hexdump-module files rotated due to volume limit	ボリューム制限に達したために、16進ダンプファイルが閉じられ、新規の16進ダンプファイルが作成された合計回数。
Hexdump-module files rotated due to time limit	時間制限に達したために、16進ダンプファイルが閉じられ、新規の16進ダンプファイルが作成された合計回数。

## show { hexdump-module | cdr } file-space-usage

フィールド	説明
Hexdump-module files rotated due to tariff-time	タリフ時間に達したため、16進ダンプファイルが閉じられ、新しい16進ダンプファイルが作成された合計回数。
Hexdump-module files rotated due to records limit	レコード制限に達したために、16進ダンプファイルが閉じられ、新規の16進ダンプファイルが作成された合計回数。
Hexdump-module file rotation failures	16進ダンプファイルの循環が失敗した合計回数。
Hexdump-module files deleted	16進ダンプファイルが削除された合計回数。
Hexdump-module records deleted	16進ダンプレコードが削除された合計回数。
Hexdump-module records received	16進ダンプレコードを受信した合計回数。
Current open Hexdump-module files	現在開いている16進ダンプファイルの合計数。
Time of last Hexdump-module file deletion	最後に16進ダンプファイルが削除された時刻。

## show { hexdump-module | cdr } file-space-usage

この機能をサポートする `show { hexdump-module | cdr } file-space-usage` コマンドの出力では、次のフィールドを使用できます。

```
CDRMOD Instance Id: 2
  Hexdump-module File Storage LIMIT      : 33554432 bytes
  Hexdump-module File Storage USAGE     : 196608 bytes
  Percentage of Hexdump-module file store usage : 0.585938
```

表 18: `show { hexdump-module | cdr } file-space-usage` コマンド出力の説明

フィールド	説明
CDRMOD インスタンス ID	統計情報が収集される CDRMOD のインスタンス ID を示します。
Hexdump-module File Storage LIMIT	16進ダンプファイルに使用できる最大ストレージ容量 (バイト単位) を示します。
Hexdump-module File Storage USAGE	16進ダンプファイルに使用する総ストレージ容量 (バイト単位) を示します。
Percentage of Hexdump-module file store usage	16進ダンプファイルに使用する総ストレージ容量のパーセンテージを示します。

## show hexdump-module statistics

この機能をサポートする **show hexdump-module statistics** コマンドの出力では、次のフィールドを使用できます。

```
Hexdump-module-Record file Statistics:
-----
CDRMOD Instance Id: 2
Hexdump-module files rotated: 0
Hexdump-module files rotated due to volume limit: 0
Hexdump-module files rotated due to time limit: 0
Hexdump-module files rotated due to tariff-time: 0
Hexdump-module files rotated due to records limit: 0
Hexdump-module file rotation failures: 0
Hexdump-module files deleted: 0
Hexdump-module records deleted: 0
Hexdump-module records received: 0
Current open Hexdump-module files: 0
Time of last Hexdump-module file deletion: 0

Hexdump-module PUSH Statistics:
-----
Successful File Transfers : 0
Failed File Transfers : 0
Num of times PUSH initiated : 0
Num of times PUSH Failed : 0
Num of times PUSH cancelled
    due to HD failure : 0
Num of periodic PUSH : 0
Num of manual PUSH : 0
Current status of PUSH : Not Running
Last completed PUSH time : N/A

Primary Server Statistics:
Successful File Transfers : 0
Failed File Transfers : 0
Num of times PUSH initiated : 0
Num of times PUSH Failed : 0
Num of periodic PUSH : 0
Num of manual PUSH : 0
Current status of PUSH : Not Running
Last completed PUSH time : N/A

Secondary Server Statistics:
Successful File Transfers : 0
Failed File Transfers : 0
Num of times PUSH initiated : 0
Num of times PUSH Failed : 0
Num of periodic PUSH : 0
Num of manual PUSH : 0
Current status of PUSH : Not Running
Last completed PUSH time : N/A
```



---

**重要** 16進ダンプモジュールの統計情報をクリアし、リセットするには、Exec モードで **clear hexdump-module statistics** コマンドを使用します。

---

表 19: show hexdump-module statistics コマンド出力の説明

フィールド	説明
<b>Hexdump-module-Record file Statistics :</b>	
CDRMOD インスタンス ID	統計情報が収集される CDRMOD のインスタンス ID を示します。
Hexdump-module files rotated	16 進ダンプファイルが閉じられ、新しい 16 進ダンプファイルが作成された合計回数。
Hexdump-module files rotated due to volume limit	ボリューム制限に達したために、16 進ダンプファイルが閉じられ、新規の 16 進ダンプファイルが作成された合計回数。
Hexdump-module files rotated due to time limit	時間制限に達したために、16 進ダンプファイルが閉じられ、新規の 16 進ダンプファイルが作成された合計回数。
Hexdump-module files rotated due to tariff-time	タリフ時間に達したため、16 進ダンプファイルが閉じられ、新しい 16 進ダンプファイルが作成された合計回数。
Hexdump-module files rotated due to records limit	レコード制限に達したために、16 進ダンプファイルが閉じられ、新規の 16 進ダンプファイルが作成された合計回数。
Hexdump-module file rotation failures	16 進ダンプファイルの循環が失敗した合計回数。
Hexdump-module files deleted	16 進ダンプファイルが削除された合計回数。
Hexdump-module records deleted	16 進ダンプレコードが削除された合計回数。
Hexdump-module records received	16 進ダンプレコードを受信した合計回数。
Current open Hexdump-module files	現在開いている 16 進ダンプファイルの合計数。
Time of last Hexdump-module file deletion	最後に 16 進ダンプファイルが削除された時刻。
<b>Hexdump-module PUSH Statistics :</b>	
Successful File Transfers	正常に転送された 16 進ダンプファイルの合計数。
Failed File Transfers	転送に失敗した 16 進ダンプファイルの合計数。
Num of times PUSH initiated	プッシュ操作が開始された合計回数。
Num of times PUSH Failed	プッシュ操作が失敗した合計回数。
Num of times PUSH cancelled due to HD failure	ハードディスクの障害が原因でプッシュ操作が失敗した合計回数。



フィールド	説明
Num of periodic PUSH	プッシュ操作が実行された周期の合計数。
Num of manual PUSH	プッシュ操作が手動で実行された合計回数。
Current status of PUSH	プッシュ操作が現在実行されているかどうか。
Last completed PUSH time	最後のプッシュ操作が完了した時刻。
<b>Primary Server Statistics :</b>	
Successful File Transfers	プライマリストレージサーバーに正常に転送された 16 進ダンプファイルの合計数。
Failed File Transfers	プライマリストレージサーバーへの転送に失敗した 16 進ダンプファイルの合計数。
Num of times PUSH initiated	プライマリストレージサーバーに 16 進ダンプファイルを送送するためのプッシュ操作を開始した合計回数。
Num of times PUSH Failed	プライマリストレージサーバーに 16 進ダンプファイルを送送するためのプッシュ操作に失敗した合計回数。
Num of periodic PUSH	プライマリストレージサーバーに対してプッシュ操作を実行した周期の合計数。
Num of manual PUSH	プライマリストレージサーバーに対してプッシュ操作を手動で実行した合計回数。
Current status of PUSH	プライマリストレージサーバーに対するプッシュ操作が現在実行されているかどうか。
Last completed PUSH time	プライマリストレージサーバーに対する最後のプッシュ操作が完了した時刻。
<b>Secondary Server Statistics :</b>	
Successful File Transfers	セカンダリストレージサーバーに正常に転送された 16 進ダンプファイルの合計数。
Failed File Transfers	セカンダリストレージサーバーへの転送に失敗した 16 進ダンプファイルの合計数。
Num of times PUSH initiated	セカンダリストレージサーバーに 16 進ダンプファイルを送送するためのプッシュ操作を開始した合計回数。
Num of times PUSH Failed	セカンダリストレージサーバーに 16 進ダンプファイルを送送するためのプッシュ操作に失敗した合計回数。

## show hexdump-module statistics

フィールド	説明
Num of periodic PUSH	セカンダリストレージサーバーに対してプッシュ操作を実行した周期の合計数。
Num of manual PUSH	セカンダリストレージサーバーに対してプッシュ操作を手動で実行した合計回数。
Current status of PUSH	セカンダリストレージサーバーに対するプッシュ操作が現在実行されているかどうか。
Last completed PUSH time	セカンダリストレージサーバーに対する最後のプッシュ操作が完了した時刻。



## 第 19 章

# システム リカバリ

この章では、電源オフサイクルの後に再起動が失敗に終わった場合、または **reload** コマンドの後に通常のブートシーケンスが中断した場合に、システムを回復する方法について説明します。



**注意** このシステムリカバリプロセスでは、既存のフローをドロップし、起動間隔中にトラフィックが処理されないようにすることによって、サブスクリバサービスを中断します。これは、緊急措置としてのみ開始する必要があります。

この章は、次の項で構成されています。

- [前提条件 \(275 ページ\)](#)
- [ブート CLI へのアクセス \(276 ページ\)](#)
- [起動できないシステムからの復旧 \(277 ページ\)](#)

## 前提条件

再起動の失敗から回復するには、ハイパーバイザコンソールを介した VPC-SI または VPC-DI CF VM へのアクセスが許可されている必要があります。またハイパーバイザにアクセス可能な StarOS .bin および .iso イメージファイルの破損していないコピーが必要です。

## コンソールアクセス

ブートリカバリシーケンスは、ハイパーバイザコンソールからのみ実行できます。

## 起動イメージ

SYSLINUX ブートローダーでは、システムを起動する起動イメージの優先順位を指定できます。ソフトウェアのアップデート後に VPC VM のリロードに失敗した場合は、以前に保存したイメージから起動を開始できます。

システムリカバリのプロセスによって、システムの起動元となる StarOS 起動イメージの場所のパス名を入力するように求めるプロンプトが表示されます。デフォルトでは、boot コマンドはタイムアウトし、デフォルトのコンフィギュレーションファイルを使用して、フラッシュメモリから優先順位の最も高いイメージのリロードを試みます。

StarOS のソフトウェアは、単一のバイナリファイル（ファイル拡張子 **.bin**）として提供され、システム全体の単一インスタンスとしてロードされます。

- イメージファイル名は、プラットフォームタイプとリリース番号で識別されます。Format = *platform-release\_number.bin*.

マルチブートの優先順位が提供され、それぞれが起動イメージ (.bin) とコンフィギュレーションファイルで構成されます。起動優先順位の番号が最も小さいものが、起動のたびに自動的に起動されます。ただし、スタートアップ時に SYSLINUX "boot:" プロンプトで番号を入力することにより、異なる優先順位を手動で起動できます。



(注) VPC VM は、ネットワークからの起動をサポートしていません。ローカル vHDD からのみ起動できます。

起動スタックのエントリと優先順位付けの詳細については、「ソフトウェア管理の運用」の章の「起動スタックの設定」の項を参照してください。

## ブート CLI へのアクセス

ブート CLI にアクセスするには、進行中のリロード (リブート) シーケンスを中断する必要があります。



**注意** このシステムリカバリプロセスでは、既存のフローをドロップし、起動間隔中にトラフィックが処理されないようにすることによって、サブスクリバサービスを中断します。これは、緊急措置としてのみ開始する必要があります。

## 再起動の開始

リロードは、ハイパーバイザ GUI を介して VM を再起動することによって開始されます。これにより、SYSLINUX ブートローダーが自動的に起動します。

ブートシーケンスでは、プロセスをステップ実行する際に、コンソールにメッセージが表示されます。

*boot:* プロンプトで、目的のブートファイルの優先順位番号を入力します。

## Enter CLI Mode

起動プロンプトが表示されたら、**cli**と入力して起動リカバリのCLIにアクセスします。CLIプロンプトが次のように変更されます。

```
8/0:boot>cli
8/0:cli>
```

## boot コマンドシンタックス

boot recovery コマンドのシンタックスは次のとおりです。

```
boot [ -show | -priority=* | -config=* | -noconfig ] { bootfile_URL }
```

このコマンドのオプションは、次のようなものがあります。

- **-show** : 現在の起動設定を表示します。
- **-priority = \*** : 必要な起動スタックの優先順位 (\*) を選択します。
- **-config = \*** : デフォルトファイルでない場合は、必要な設定ファイル名 (\*) を入力します。
- **-noconfig** : 設定ファイルを使用せずに起動します。

**bootfile\_URL** は、StarOS 起動イメージファイルの場所の URL です。システムを起動する StarOS .bin ファイルのパスとファイル名を指定します。

URL は、管理カードに接続されているメモリデバイス上のローカルファイル (flash) や外部ファイルを参照できます。URL は次の形式で入力する必要があります。

```
{ /flash | /pcmcial | /usb1 }/filename
```

## 起動できないシステムからの復旧

VPCが起動できなくなった場合 (すべてのイメージを削除したり、起動優先順位を誤って設定したなど)、インストーラ ISO (.ssi.iso ファイル) を起動するか、第2のオプション (recover) を選択することで、復旧できます。このオプションでは、起動可能な新しい .bin ファイルをインストールし、新しい起動優先順位リストを作成します。vHDDは再フォーマットされません (これを実行するにはオプション1を選択)。そのため、設定ファイルが保持されます。





## 第 20 章

# アクセスコントロールリスト

この章では、アクセスコントロールリストのシステムサポートについて説明し、それらの設定方法を示します。製品アドミニストレーションガイドには、システム上での基本サービスの設定例と手順が示されています。次に説明する手順を使用する前に、サービスモデルに最適な設定例を選択する必要があります。



**重要** ACL を設定するためのライセンスは必要ありません。ただし、設定されている ACL の数は、パフォーマンスに大きく影響する可能性があります。



**重要** すべてのコマンドとキーワード/変数を使用できるわけではありません。可用性はプラットフォームのタイプによって異なります。

この章は、次の項で構成されています。

- [概要 \(279 ページ\)](#)
- [ACL の概要 \(280 ページ\)](#)
- [システム上での ACL の設定 \(282 ページ\)](#)
- [IP ACL の適用 \(285 ページ\)](#)

## 概要

IP アクセスリスト (一般にアクセスコントロールリスト (ACL) と呼ばれる) は、システムに出入りするパケットのフローを制御します。これらはコンテキストごとに設定され、フィルタ基準に一致するパケットに対して実行されるアクションを制御する「ルール」(ACL ルール) またはフィルタで構成されます。設定が完了すると、ACL を次のいずれかに適用できます。

- 個別のインターフェイス
- コンテキストによって促進されるすべてのトラフィック (ポリシー ACL と呼ばれる)
- 個々のサブスクライバ

- 特定のコンテキストによって促進されるすべてのサブスクライバセッション

IPv4 と IPv6 のアクセスルートに個別の ACL を作成できます。

## ACL の概要

この項では、システム上の ACL に関する 2 つの重要な側面について説明します。

- [ルール \(280 ページ\)](#)
- [ルールの順序 \(282 ページ\)](#)



---

**重要** 完全なコマンドシンタックスについては、『*Command Line Interface Reference*』の「*ACL Configuration Mode Commands*」および「*IPv6 ACL Configuration Mode Commands*」の章を参照してください。

---

## ルール

単一の ACL は、1 つ以上の ACL ルールで構成されます。各ルールは、パケットが特定の基準に一致した場合に、特定のアクションを実行するように設定されたフィルタです。ACL ごとに最大 256 のルールを設定できます。



---

**重要** ルールなしで構成されている設定済み ACL は、「deny any」ルールを示します。deny アクションおよび any 基準については、このセクションの後半で説明します。これは、空の ACL のデフォルトの動作です。

---

各ルールは、指定された基準にパケットが一致した場合に、実行するアクションを指定します。この項では、システムでサポートされているルールアクションと基準について説明します。

## アクション

ACL は、指定された基準に一致するパケットに対して、次のいずれかのアクションを実行できるように指定します。

- **許可**：パケットは受け入れられ、処理されます。
- **拒否**：パケットは拒否されます。
- **リダイレクト**：パケットは特定のシステムインターフェイスまたは処理のために指定されたコンテキストを介して、指定されたネクストホップアドレスに転送されます。





---

**重要** リダイレクトルールは、特定のサブスクライバ、特定のコンテキストによって促進されるすべてのサブスクライバ、または UMTS サブスクライバ用の APN に適用される ACL の場合は無視されません。

---

## 基準

各 ACL は、パケットの比較基準を指定する 1 つまたは複数のルールで構成されます。

サポートされる基準は次のとおりです。

- **Any** : すべてのパケットをフィルタ処理します。
- **Host** : 送信元ホストの IP アドレスに基づいてパケットをフィルタ処理します。
- **ICMP** : Internet Control Message Protocol (ICMP) のパケットをフィルタ処理します。
- **IP** : Internet Protocol (IP) パケットをフィルタ処理します。
- **Source IP Address** : 1 つ以上の送信元 IP アドレスに基づいてパケットをフィルタ処理します。
- **TCP** : Transport Control Protocol (TCP) パケットをフィルタ処理します。
- **UDP** : User Datagram Protocol (UDP) パケットをフィルタ処理します。

上記の基準それぞれについては、以降の項で詳しく説明します。



---

**重要** 次の項では、基本的な ACL ルールのシンタックスについて説明します。コマンドシンタックスの詳細については、『*Command Line Interface Reference*』の「*ACL Configuration Mode Commands*」の章と「*IPv6 ACL Configuration Mode Commands*」の章を参照してください。

---

- **Any** : すべてのパケットにルールが適用されます。
- **Host** : IP アドレスによって決定される特定のホストにルールが適用されます。
- **ICMP** : 特定の Internet Control Message Protocol (ICMP) パケット、タイプ、またはコードにルールが適用されます。ICMP のタイプとコードの定義については、[www.iana.org](http://www.iana.org) (RFC 3232) を参照してください。
- **IP** : 特定の Internet Protocol (IP) パケットまたはフラグメントにルールが適用されます。
- **IP Packet Size Identification Algorithm** : 転送時にフラグメンテーションの特定の Internet Protocol (IP) パケット ID にルールが適用されます。

この設定は、サブスクライバパケットがカプセル化されている場合 (モバイル IP やその他のトンネリングカプセル化など) に、システムで使用される「IP ID フィールド」割り当てアルゴリズムに関連しています。システム内では、サブスクライバパケットのカプセ

ル化は分散型の方法で行われ、16 ビットの IP ID 空間が分割されてカプセル化を行う各エンティティに分散されるため、カプセル化時に一意の IP ID 値を IP ヘッダーに割り当てることができます。

この分散型の IP ID 空間は小規模であるため、ゼロ以外の一位の ID は、転送時にフラグメント化される可能性があるパケットのみに割り当てられます。これは、IP ID フィールドは、フラグメント化されたパケットのリアセンブルにのみ使用されるためです。IP パケットの合計サイズは、そのパケットがフラグメント化される可能性を判断するために使用されます。

- **Source IP Address** : 特定の送信元アドレスまたは送信元アドレスのグループから発信される特定のパケットにルールが適用されます。
- **TCP** : 任意の Transport Control Protocol (TCP) トラフィックにルールが適用され、送信元/接続先の IP アドレス、特定のポート番号、またはポート番号のグループの任意の組み合わせでフィルタ処理されます。TCP ポート番号の定義については、[www.iana.org](http://www.iana.org) を参照してください。
- **UDP** : 任意の User Datagram Protocol (UDP) トラフィックにルールが適用され、送信元/接続先の IP アドレス、特定のポート番号、またはポート番号のグループの任意の組み合わせでフィルタ処理されます。UDP ポート番号の定義については、[www.iana.org](http://www.iana.org) を参照してください。

## ルールの順序

複数のルールで 1 つの ACL を構成できます。各パケットは、一致が見つかるまで、各 ACL ルールを入力した順序で比較されます。一致が特定されると、後続のすべてのルールは無視されます。

追加のルールを既存の ACL に追加し、次のいずれかのオプションを使用して適切に順序付けることができます。

- Before
- After

これらの配置オプションを使用するには、ACL 内に既存のルールを指定し、次のフローに示すように新しいルールを設定する必要があります。

```
[ before | after ] { existing_rule }
```

## システム上での ACL の設定

ここでは、ACL の設定方法について説明します。



**重要** この項では、システムでアクセスコントロールリストを設定するための最小の命令セットについて説明します。追加のパラメータとオプションを設定するコマンドの詳細については、『*Command Line Interface Reference*』の「*ACL Configuration Mode Commands*」の章と「*IPv6 ACL Configuration Mode Commands*」の章を参照してください。

サブスクリバにアクセスコントロールリストの機能を提供するようにシステムを設定するには、次の手順を実行します。

- ステップ1 の設定例に従って、アクセスコントロールリストを作成します。[ACLの作成 \(283 ページ\)](#)
- ステップ2 の設定例に従って、ACLリスト内のアクションのルールと基準を指定します。[サブスクリバトラフィックのアクションと基準の設定 \(283 ページ\)](#)
- ステップ3 オプションです。システムには、コンテキストへのすべてのパケットのデフォルトフィルタとして機能する「未定義」ACLが用意されています。デフォルトのアクションは「`permit all`」です。の設定例に従って、「未定義」ACLのデフォルト設定を変更します。[未定義のACLの設定 \(284 ページ\)](#)
- ステップ4 の手順に従って、ACLの設定を確認します。[ACL設定の確認 \(284 ページ\)](#)
- ステップ5 Execモードの**`save configuration`** コマンドを使用して、設定をフラッシュメモリ、外部メモリデバイス、またはネットワークの場所に保存します。詳細については、「設定の確認と保存」の章を参照してください。

## ACLの作成

ACLを作成するには、システムCLIのExecモードから次のコマンドシーケンスを入力します。

```
configure
context acl_ctxt_name [ -noconfirm ]
  { ip | ipv6 } access-list acl_list_name
end
```

注：

- コンテキストごとに設定できるACLの最大数は、VPN Manager ソフトウェアタスクで使用可能なメモリ量によって制限されます。通常は、最大 200 未満です。

## サブスクリバトラフィックのアクションと基準の設定

サブスクリバトラフィックを拒否/許可するルールを作成し、アクションの前後にルールを適用するには、システムCLIのExecモードから次のコマンドシーケンスを入力します。

```
configure
context acl_ctxt_name [ -noconfirm ]
  { ip | ipv6 } access-list acl_list_name
    deny { ip_address | any | host | icmp | ip | log | tcp | udp }
```

```

permit { ip_address | any | host | icmp | ip | log | tcp | udp }
after { deny | permit | readdress | redirect }
before { deny | permit | readdress | redirect }
end

```

注：



**注意** ACLで指定されていない限り、システムは「deny any」ルールを適用しません。この動作は、ACLの最後に「deny any」ルールを追加することによって変更できます。

- ACLごとに設定できるルールの最大数は、ACLがどのように使用されるかによって異なります。詳細については、「エンジニアリングルール」の章を参照してください。
- ACLを構成するルールを設定するには、[アクション](#)と[基準](#)に表示される情報を使用します。詳細については、『*Command Line Interface Reference*』の「*Acl configuration mode commands*」および「*IPv6 acl configuration mode commands*」の章を参照してください。

## 未定義の ACL の設定

前述のように、システムでは、適用されている ACL が存在しない場合に、パケットのフィルタリングに「未定義」の ACL メカニズムが使用されます。このシナリオは、設定プロセス中に ACL 名が誤って入力されたなどの誤設定が原因である可能性があります。

このような状況に備えて、システムには、コンテキストへのすべてのパケットのデフォルトフィルタとして機能する「未定義」の ACL が用意されています。デフォルトのアクションは「permit all」です。

確認できない ACL のデフォルトの動作を変更するには、次の設定を使用します。

```

configure
context acl_ctxt_name [-noconfirm]
  access-list undefined { deny-all | permit-all }
end

```

注：

- コンテキスト名は、変更する「未定義」の ACL を含むコンテキストの名前です。詳細については、『*Command Line Interface Reference*』の「*Context Configuration Mode Commands*」の章を参照してください。

## ACL 設定の確認

ACL の設定を確認するには、Exec モードの `show { ip | ipv6 } access-list` コマンドを入力します。

次に、このコマンドの出力例を示します。この例では、`acl_1` という名前の ACL が設定されています。

```

ip access list acl_1
  deny host 10.2.3.4

```

```
deny ip any host 10.2.3.4
permit any 10.2.4.4
1 ip access-lists are configured.
```

## IP ACL の適用

ACL を設定した後、有効にするには、ACL を適用する必要があります。



**重要** これらの手順を開始する前に、システム上での ACL の設定 (282 ページ) の手順に従って、すべての ACL を設定し、検証する必要があります。また、次に示す手順では、サブスクライバが事前に設定されていることも前提としています。

前述のように、次のいずれかに ACL を適用できます。

- 個々のインターフェイスへの ACL の適用 (287 ページ)
- コンテキスト内のすべてのトラフィックへの ACL の適用 (289 ページ) (ポリシー ACL として知られている)
- 個々のサブスクライバへの ACL の適用 (290 ページ)
- 複数のサブスクライバへの単一 ACL の適用 (295 ページ)
- 複数のサブスクライバへの単一 ACL の適用 (295 ページ) (3GPP サブスクライバの場合のみ)



**重要** ACL は、適用先のサブスクライバやインターフェイス内の同じコンテキストで設定する必要があります。同様に、コンテキストに適用される ACL は、そのコンテキストで設定する必要があります。

ACL が単一のコンテキスト内の複数のレベルで適用される場合 (ACL がコンテキスト内のインターフェイスに適用され、コンテキスト全体に別の ACL が適用されるなど)、次の図と表に示すように処理されます。

図 2: ACL の処理順序

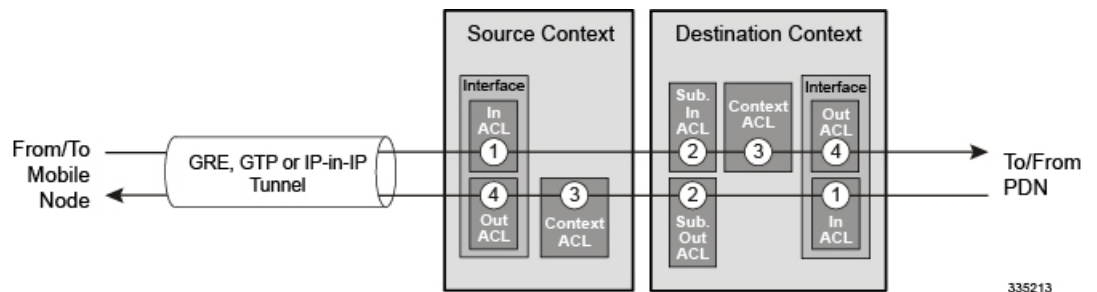


表 20: ACL の処理順序の説明

モバイルノードからパケットデータネットワークに着信するパケット (左から右)

順序	説明
1	送信元コンテキストの受信インターフェイス用に設定されたインバウンド ACL は、トンネリングされたデータ（外部 IP ヘッダーなど）に適用されます。その後、パケットは接続先コンテキストに転送されます。
2	サブスライバに対して設定されたインバウンド ACL（特定のサブスライバまたはコンテキストによって促進された任意のサブスライバのいずれか）が適用されます。
3	接続先コンテキストで設定されたコンテキスト ACL（ポリシー ACL）は、転送前に適用されます。
4	パケットが転送される接続先コンテキストのインターフェイスに設定されたアウトバウンド ACL が適用されます。

パケットデータネットワークからモバイルノードに着信するパケット（右から左）

順序	説明
1	接続先コンテキストで設定された受信インターフェイス用に、設定されたインバウンド ACL が適用されます。
2	サブスライバに対して設定されたアウトバウンド ACL（特定のサブスライバまたはコンテキストによって促進された任意のサブスライバのいずれか）が適用されます。その後、パケットは送信元コンテキストに転送されます。
3	送信元コンテキストで設定されたコンテキスト ACL（ポリシー ACL）は、転送前に適用されます。
4	パケットが転送される送信元コンテキストのインターフェイスに設定されたアウトバウンド ACL が、トンネリングされたデータ（外部 IP ヘッダーなど）に適用されます。

設定されていない IP ACL が適用されている場合（たとえば、適用された ACL の名前が誤って設定されている場合）、システムはパケットのフィルタリングに「未定義」の ACL メカニズムを使用します。

この項では、ACL の適用および「未定義」の ACL の設定に関する情報と手順を示します。

## インターフェイスへの ACL の適用

ACL をインターフェイスに適用するには、次の設定を使用します。

```
configure
  context acl_ctxt_name [ -noconfirm ]
    interface interface_name
      { ip | ipv6 } access-group acl_list_name { in | out } [ preference
    ]
  end
```

注：

- コンテキスト名は、ACL を適用するインターフェイスを含む ACL コンテキストの名前です。
- 適用する ACL は、このコマンドで指定されたコンテキストで設定する必要があります。
- ACL 内に設定されているルール数が、そのインターフェイスの 128 ルールの制限を超えていない場合、最大 16 の ACL をグループに適用できます。

## 個々のインターフェイスへの ACL の適用

この項では、システムに設定されている個々のインターフェイスに 1 つまたは複数の ACL を適用するための情報と手順について説明します。



**重要** この項では、システム上のインターフェイスに ACL リストを適用するための最小の命令セットについて説明します。追加のパラメータとオプションを設定するコマンドの詳細については、『*Command Line Interface Reference*』の「*Ethernet Interface Configuration Mode Commands*」の章を参照してください。

サブスクライバに ACL ファシリティを提供するようにシステムを設定するには、次のようにします。

- ステップ 1** 設定例に従って、設定されたアクセスコントロールリストを適用します。 [インターフェイスへの ACL の適用 \(286 ページ\)](#)
- ステップ 2** ステップに従って、ACL がインターフェイスに適切に適用されていることを確認します。 [インターフェイス上の ACL 設定の確認 \(287 ページ\)](#)
- ステップ 3** Exec モードの **save configuration** コマンドを使用して、設定をフラッシュメモリ、外部メモリデバイス、またはネットワークの場所に保存します。詳細については、「設定の確認と保存」の章を参照してください。

## インターフェイス上の ACL 設定の確認

この項では、ACL 設定を確認する方法について説明します。

Exec モードで、次のコマンドを入力します。

```
[local]host_name# show configuration context context_name
```

`context_name` は、ACL が適用されたインターフェイスを含むコンテキストの名前です。

このコマンドの出力には、コンテキスト全体の設定が表示されます。インターフェイス設定に関するコマンドの出力を確認します。このコマンドは、この手順を使用して適用された ACL を表示します。

```
configure
  context context_name
    ip access-list acl_name
      deny host ip_address
      deny ip any host ip_address
```

```

    exit
ip access-group access_group_name
service-redundancy-protocol
exit
interface interface_name
    ip address ip_address/mask
    exit
subscriber default
exit
aaa group default
exit
gtpp group default
end

```

## コンテキストへの ACL の適用

ACL をコンテキストに適用するには、次の設定を使用します。

### configure

```

context acl_ctxt_name [ -noconfirm ]
    { ip | ipv6 } access-group acl_list_name [ in | out ] [ preference ]
end

```

注：

- コンテキスト名は、ACL を適用するインターフェイスを含む ACL コンテキストの名前です。
- コンテキストレベルの ACL は、発信パケットに適用されます。これは、フロー一致基準が失敗して再度転送された場合にも、着信パケットに適用されます。

**in** キーワードと **out** キーワードは廃止されており、後方互換性のためにのみ存在します。

コンテキスト ACL は、次の場合に適用されます。

- 外部ソースへの発信パケット。
- 失敗したフローが一致し、再度転送される着信パケット。この場合、コンテキスト ACL が最初に適用され、通過した場合のみパケットが転送されます。

転送中に、ACL ルールが宛先アドレスとしてループバックアドレスとして追加されると、コンテキスト ACL も適用されます。これは、StarOS がカーネル宛でのパケットをフォワーディング ルックアップで処理するためです。ACL ルールを着信パケットに適用するには、コンテキスト ACL の代わりにインターフェイス ACL を使用する必要があります。

- 適用する ACL は、このコマンドで指定されたコンテキストで設定する必要があります。
- ACL 内で設定されているルール数が、そのインターフェイスの 256 ルールの制限を超えていない場合、最大 16 の ACL をグループに適用できます。



## コンテキスト内のすべてのトラフィックへの ACL の適用

この項では、システム上の特定のコンテキスト内で設定されたコンテキストに1つ以上の ACL を適用する手順について説明します。適用される ACL (ポリシー ACL と呼ばれる) には、コンテキストによって容易になるすべてのトラフィックに適用されるルールが含まれています。



**重要** この項では、コンテキスト内のすべてのトラフィックに ACL リストを適用するための最小の命令セットについて説明します。追加のパラメータとオプションを設定するコマンドの詳細については、『*Command Line Interface Reference*』の「*Context Configuration Mode Commands*」の章を参照してください。

サブスクリバにアクセスコントロールリストの機能を提供するようにシステムを設定するには、次の手順を実行します。

- ステップ 1** の説明に従って、設定された ACL を適用します。 [コンテキストへの ACL の適用 \(288 ページ\)](#)
- ステップ 2** の説明に従って、ACL がインターフェイスに適切に適用されていることを確認します。 [コンテキストでの ACL 設定の確認 \(289 ページ\)](#)
- ステップ 3** Exec モードの **save configuration** コマンドを使用して、設定をフラッシュメモリ、外部メモリデバイス、またはネットワークの場所に保存します。詳細については、「設定の確認と保存」の章を参照してください。

## コンテキストでの ACL 設定の確認

ACL の設定を確認するには :

Exec モードで次のコマンドを入力して、ACL リストが適切に適用されていることを確認します。

```
[local]host_name# show configuration context context_name
```

*context\_name* は、ACL が適用されたコンテキストの名前です。

このコマンドの出力には、コンテキスト全体の設定が表示されます。インターフェイス設定に関するコマンドの出力を確認します。このコマンドは、この手順を使用して適用された ACL を表示します。

```
configure
context context_name
  ip access-list acl_name
  deny host ip_address
  deny ip any host ip_address
  exit
  ip access-group access_group_name
  service-redundancy-protocol
  exit
  interface interface_name
  ip address ip_address/mask
  exit
subscriber default
exit
aaa group default
```

```
exit
gtpp group default
end
```

## RADIUS ベースのサブスクリバにおける ACL の適用

IP ACL は、プロファイル内の属性を介してサブスクリバに適用されます。サブスクリバプロファイルは、システム上にローカルで設定することも、RADIUS サーバー上にリモートで設定することもできます。

ACL を RADIUS ベースのサブスクリバに適用するには、**フィルタ ID** 属性を使用します。

この属性の詳細については、『*AAA Interface Administration and Reference*』を参照してください。

この項では、プロファイルがシステム上でローカルに設定されている個々のサブスクリバに ACL を適用するための情報と手順について説明します。



**重要** この項では、コンテキスト内のすべてのトラフィックに ACL リストを適用するための最小の命令セットについて説明します。追加のパラメータとオプションを設定するコマンドの詳細については、『*Command Line Interface Reference*』の「*Subscriber Configuration Mode Commands*」の章を参照してください。

サブスクリバにアクセスコントロールリストの機能を提供するようにシステムを設定するには、次の手順を実行します。

- ステップ 1** 設定例に従って、設定されたアクセスコントロールリストを適用します。 [個々のサブスクリバへの ACL の適用 \(290 ページ\)](#)
- ステップ 2** ステップに従って、ACL がインターフェイスに適切に適用されていることを確認します。 [個々のサブスクリバへの ACL 設定の確認 \(291 ページ\)](#)
- ステップ 3** Exec モードの **save configuration** コマンドを使用して、設定をフラッシュメモリ、外部メモリデバイス、またはネットワークの場所に保存します。詳細については、「[設定の確認と保存](#)」の章を参照してください。

## 個々のサブスクリバへの ACL の適用

ACL を個々のサブスクリバに適用するには、次の設定を使用します。

```
configure
context acl_ctxt_name [ -noconfirm ]
subscriber name subs_name
{ ip | ipv6 } access-group acl_list_name [ in | out ]
end
```

注：

- コンテキスト名は、ACL を適用するインターフェイスを含む ACL コンテキストの名前です。
- **in** キーワードも **out** キーワードも指定しなかった場合、ACL はインバウンドとアウトバウンドのすべてのパケットに適用されます。
- 適用する ACL は、このコマンドで指定されたコンテキストで設定する必要があります。
- ACL 内で設定されているルール数が、そのインターフェイスの 128 ルールの制限を超えていない場合、最大 8 つの ACL をグループに適用できます。

## 個々のサブスクリイバへの ACL 設定の確認

次の手順は、ACL の設定を確認するために使用されます。

---

Exec モードで次のコマンドを入力して、ACL リストが適切に適用されていることを確認します。

```
[local]host_name# show configuration context context_name
```

*context\_name* は、ACL が適用されたサブスクリイバ *subs1* を含むコンテキストの名前です。

このコマンドの出力には、コンテキスト全体の設定が表示されます。インターフェイス設定に関するコマンドの出力を確認します。このコマンドは、この手順を使用して適用された ACL を表示します。

```
configure
  context context_name
    ip access-list acl_name
      deny host ip_address
      deny ip any host ip_address
    exit
  ip access-group access_group_name
  service-redundancy-protocol
  exit
  interface interface
    ip address ip_address/mask
  exit
  subscriber default
  exit
  subscriber name subscriber_name
    ip access-group access_group_name in
    ip access-group access_group_name out
  exit
  aaa group default
  exit
  gtp group default
  exit
  content-filtering server-group cfsq_name
    response-timeout response_timeout
    connection retry-timeout retry_timeout
  end
```

---

## default というサブスクリイバへの ACL の適用

この項では、*default* という名前のサブスクリイバに ACL を適用する方法について説明します。



**重要** この項では、コンテキスト内のすべてのトラフィックに ACL リストを適用するための最小の命令セットについて説明します。追加のパラメータとオプションを設定するコマンドの詳細については、『*Command Line Interface Reference*』の「*Subscriber Configuration Mode Commands*」を参照してください。

サブスクリイバにアクセスコントロールリストの機能を提供するようにシステムを設定するには、次の手順を実行します。

- ステップ 1 設定例に従って、設定されたアクセスコントロールリストを適用します。 [default というサブスクリイバへの ACL の適用 \(292 ページ\)](#)
- ステップ 2 ステップに従って、ACL がインターフェイスに適切に適用されていることを確認します。 [default というサブスクリイバに対する ACL 設定の確認 \(292 ページ\)](#)
- ステップ 3 Exec モードの **save configuration** コマンドを使用して、設定をフラッシュメモリ、外部メモリデバイス、またはネットワークの場所に保存します。詳細については、「設定の確認と保存」の章を参照してください。

## default というサブスクリイバへの ACL の適用

*default* というサブスクリイバに ACL を適用するには、次の設定を使用します。

```
configure
context acl_ctxt_name [ -noconfirm ]
subscriber name subs_name
  { ip | ipv6 } access-group acl_list_name [ in | out ]
end
```

注：

- コンテキスト名は、ACL を適用するインターフェイスを含む ACL コンテキストの名前です。
- **in** キーワードも **out** キーワードも指定しなかった場合、ACL はインバウンドとアウトバウンドのすべてのパケットに適用されます。
- 適用する ACL は、このコマンドで指定されたコンテキストで設定する必要があります。
- ACL 内で設定されているルール数が、そのインターフェイスの 256 ルールの制限を超えていない場合、最大 16 の ACL をグループに適用できます。

## default というサブスクリイバに対する ACL 設定の確認

次の手順は、ACL の設定を確認するために使用されます。

Exec モードで次のコマンドを入力して、ACL リストが適切に適用されていることを確認します。

```
[local]host_name# show configuration context context_name
```

`context_name` は、ACL が適用された `default` というサブスクリバを含むコンテキストの名前です。

このコマンドの出力には、コンテキスト全体の設定が表示されます。インターフェイス設定に関するコマンドの出力を確認します。このコマンドは、この手順を使用して適用された ACL を表示します。

```
configure
context context_name
  ip access-list acl_name
  deny host ip_address
  deny ip any host ip_address
  exit
  ip access-group access_group_name
  service-redundancy-protocol
  exit
  interface interface
    ip address ip_address/mask
    exit
  subscriber name default
    ip access-group access_group_name in
    ip access-group access_group_name out
    exit
  aaa group default
  exit
  gtpv group default
  exit
  content-filtering server-group cfsq_name
  response-timeout response_timeout
  connection retry-timeout retry_timeout
  end
```

## サービス指定のデフォルトのサブスクリバへの ACL の適用

この項では、さまざまなシステムサービスによって「デフォルト」のプロファイルとして使用されるサブスクリバへの ACL の適用について説明し、手順を示します。



**重要** この項では、コンテキスト内のすべてのトラフィックに ACL リストを適用するための最小の命令セットについて説明します。追加のパラメータとオプションを設定するコマンドの詳細については、『*Command Line Interface Reference*』の「*Subscriber Configuration Mode Commands*」の章を参照してください。

サブスクリバにアクセスコントロールリストの機能を提供するようにシステムを設定するには、次の手順を実行します。

**ステップ 1** [default というサブスクリバへの ACL の適用 \(292 ページ\)](#) の設定例に従って、設定されたアクセスコントロールリストを適用します。

- ステップ 2** サービス指定のデフォルトのサブスクリバへの ACL 設定の確認 (294 ページ) の手順に従って、ACL がインターフェイスに正しく適用されていることを確認します。
- ステップ 3** Exec モードの **save configuration** コマンドを使用して、設定をフラッシュメモリ、外部メモリデバイス、またはネットワークの場所に保存します。詳細については、「設定の確認と保存」の章を参照してください。

## サービス指定のデフォルトのサブスクリバへの ACL の適用

ACL をサービス指定のデフォルトサブスクリバに適用するには、次の設定を使用します。

```
configure
context acl_ctxt_name [ -noconfirm ]
{ pdsn-service | fa-service | ha-service } service_name
  default subscriber svc_default_subs_name
  exit
subscriber name svc_default_subs_name
{ ip | ipv6 } access-group acl_list_name [ in | out ]
end
```

注：

- コンテキスト名は、ACL を適用するインターフェイスを含む ACL コンテキストの名前です。
- **in** キーワードも **out** キーワードも指定しなかった場合、ACL はインバウンドとアウトバウンドのすべてのパケットに適用されます。
- 適用する ACL は、このコマンドで指定されたコンテキストで設定する必要があります。
- ACL 内で設定されているルール数が、そのインターフェイスの 128 ルールの制限を超えていない場合、最大 8 つの ACL をグループに適用できます。

## サービス指定のデフォルトのサブスクリバへの ACL 設定の確認

ACL の設定を確認します。

Exec モードで次のコマンドを入力して、ACL リストが適切に適用されていることを確認します。

```
[local]host_name# show configuration context context_name
```

*context\_name* は、ACL が適用されたデフォルトのサブスクリバとともにサービスが含まれているコンテキストの名前です。

このコマンドの出力には、コンテキスト全体の設定が表示されます。インターフェイス設定に関するコマンドの出力を確認します。このコマンドは、この手順を使用して適用された ACL を表示します。

```
configure
context context_name
  ip access-list acl_name
  deny host ip_address
  deny ip any host ip_address
exit
```

```

ip access-group access_group_name
interface interface
    ip address ip_address/mask
    exit
subscriber default
exit
subscriber name subscriber_name
    ip access-group access_group_name in
    ip access-group access_group_name out
    exit
pdsn-service service_name
    default subscriber subscriber_name
    end

```

## 複数のサブスライバへの単一 ACL の適用

前の項で説明したように、IP ACL は、プロファイル内の属性を介してサブスライバに適用されます。サブスライバプロファイルは、システム上にローカルで設定することも、RADIUS サーバー上にリモートで設定することもできます。

システムには、特定の属性が個々のサブスライバのプロファイルに含まれていない場合にデフォルト値として機能するサブスライバ機能の設定が用意されています。次の表で、これらの機能について説明します。

表 21: 「デフォルト」サブスライバ属性を提供するために使用される機能

機能	Description
<i>default</i> という名前のサブスライバ	<p>システムは各コンテキスト内に <i>default</i> というサブスライバを伴ったプロファイルに、そのコンテキストに定義されたサブスライバの属性値の設定テンプレートが備わっている。</p> <p>RADIUS ベースのサブスライバプロファイルに含まれていないサブスライバ属性は、<i>default</i> というサブスライバに定義されている属性値で設定されます。</p> <p><b>注 :</b> <i>default</i> というサブスライバのプロファイルを使用してローカルに定義されているサブスライバの欠落情報を提供することはできません。</p>
<b>default subscriber</b>	このコマンドを使用すると、複数のサービスが複数のプロファイルに「default」サブスライバ情報を取得できます。

適切に設定されている場合は、上の表に記載されている機能を使用して ACL を以下に適用することができます。

- *default* というサブスライバのプロファイルに ACL を適用することで、特定のコンテキスト内で促進されたすべてのサブスライバ。
- サブスライバのプロファイルに ACL を適用した後、**default subscriber** コマンドを使用してそのサブスライバを「デフォルト」のプロファイルとして使用するよう設定することで促進されたすべてのサブスライバ。

## 複数のサブスライバへの APN を介した ACL の適用

APN を介して複数のサブスライバに ACL を適用するには、次の設定を使用します。

```
configure
context dest_context_name [-noconfirm]
  apn apn_name
    { ip | ipv6 } access-group acl_list_name [ in | out ]
  end
```

注：

- 適用する ACL は、APN の接続先コンテキスト内にある必要があります（APN が設定されているコンテキストとは異なる場合があります）。
- **in** キーワードも **out** キーワードも指定しなかった場合、ACL はインバウンドとアウトバウンドのすべてのパケットに適用されます。
- このコマンドは、1 つの ACL のみをサポートします。ただし、ACL には最大 256 のルールを設定できます。
- 各 APN に対して 4 つのアクセスグループを適用できます。次に例を示します。

```
ip access-group acl_list_name_1 in
ip access-group acl_list_name_2 out
ipv6 access-group acl_list_name_3 in
ipv6 access-group acl_list_name_4 out
```

## 複数のサブスライバへの APN を介した ACL の適用

IP ACL がプロファイル内の属性を使用してサブスライバに適用される場合、サブスライバプロファイルは、システム上にローカルで設定することも、RADIUS サーバー上にリモートで設定することもできます。

設定時間を短縮するために、代わりに ACL を GGSN サブスライバの APN テンプレートに適用することができます。設定されている場合、APN テンプレートによって促進されたサブスライバパケットには、関連付けられた ACL が適用されます。

この項では、APN テンプレートに ACL を適用する方法について説明します。



- 
- 重要** この項では、コンテキスト内のすべてのトラフィックに ACL リストを適用するための最小の命令セットについて説明します。追加のパラメータとオプションを設定するコマンドの詳細については、『*Command Line Interface Reference*』の「*Subscriber Configuration Mode Commands*」の章を参照してください。
- 

サブスライバにアクセスコントロールリストの機能を提供するようにシステムを設定するには、次の手順を実行します。



- 
- ステップ 1** 複数のサブスライバへの APN を介した ACL の適用 (296 ページ) の設定例に従って、設定されたアクセスコントロールリストを適用します。
- ステップ 2** APN への ACL 設定の確認 (297 ページ) のステップに従って、ACL がインターフェイスに適切に適用されていることを確認します。
- ステップ 3** Exec モードの **save configuration** コマンドを使用して、設定をフラッシュメモリ、外部メモリデバイス、またはネットワークの場所に保存します。詳細については、「設定の確認と保存」の章を参照してください。
- 

## APN への ACL 設定の確認

ACL の設定を確認するには：

---

Exec モードで次のコマンドを入力して、ACL リストが適切に適用されていることを確認します。

```
show configuration context context_name
```

*context\_name* は、ACL が適用されたデフォルトのサブスライバを持つ APN *apn1* を含むコンテキストの名前です。

このコマンドの出力には、コンテキスト全体の設定が表示されます。インターフェイス設定に関するコマンドの出力を確認します。このコマンドは、この手順を使用して適用された ACL を表示します。

```
configure
  context context_name
    ip access-list acl_name
      deny host ip_address
      deny ip any host ip_address
    exit
    ip access-group access_group_name
  interface interface
    ip address ip_address/mask
  exit
  subscriber default
  exit
  apn apn_name
    ip access-group access_group_name in
    ip access-group access_group_name out
  end
```

---





## 第 21 章

# 輻輳制御

この章では、輻輳制御機能について説明します。内容は次のとおりです。

- [概要 \(299 ページ\)](#)
- [輻輳制御の設定 \(300 ページ\)](#)

## 概要

輻輳制御は、システムに重い負荷がかかっている場合にパフォーマンスを低下させる可能性のある条件についてシステムをモニターします。通常、これらの条件は一時的なものであり（CPU 使用率やメモリ使用率が高い場合など）、すぐに解決されます。ただし、特定の時間間隔内にこれらの条件が継続しているか、または大量に発生した場合、システムがサブスクライバセッションにサービスを提供する能力に影響を与える可能性があります。輻輳制御は、このような条件を特定し、状況に対応するためのポリシーを呼び出すのに役に立ちます。

輻輳制御の動作は、次の設定に基づいています。

- **輻輳条件のしきい値**：しきい値は、輻輳制御が有効になっている条件を指定し、システムの状態（輻輳またはクリア）を定義するための制限を確立します。これらのしきい値は、システムに設定されている動作のしきい値と同様の方法で機能します（『*Thresholding Configuration Guide*』を参照）。主な違いは、輻輳のしきい値に達すると、サービス輻輳ポリシーと SNMP トラップ（starCongestion）が生成されることです。

しきい値の許容度は、条件をクリアするために到達する必要がある、設定されたしきい値のパーセンテージを決定します。次に、SNMP トラップ（starCongestionClear）がトリガーされます。

- **ポート使用率のしきい値**：ポート使用率のしきい値を設定した場合、システム内のすべてのポートの平均使用率が指定されたしきい値に達すると、輻輳制御が有効になります。
- **ポート固有のしきい値**：ポート固有のしきい値を設定した場合、個々のポート固有のしきい値に到達すると、輻輳制御はシステム全体で有効になります。

- **サービス輻輳ポリシー**：輻輳ポリシーは、サービスごとに設定できます。これらのポリシーは、輻輳状態のしきい値を超えたことをシステムが検出したときにサービスがどのように応答するかを指示します。



**重要** この項では、輻輳制御を設定するための最小の命令セットについて説明します。追加のインターフェイスまたはポートのプロパティを設定するコマンドは、『*Command Line Interface Reference*』の「*Subscriber Configuration Mode*」で提供されます。輻輳制御に関する追加の設定情報については、常にこのプラットフォームで実行されているすべてのライセンス製品のアドミニストレーションガイドを参照してください。輻輳制御機能は、製品および StarOS のバージョンによって異なります。

MME では、3 つのレベルの輻輳制御しきい値がサポートされています（クリティカル、メジャー、マイナー）。デフォルトでは、他の製品に対しサポートされているのは重大しきい値のみです。SNMP トラップは、メジャー輻輳制御しきい値およびマイナー輻輳制御しきい値もサポートしています。一連の **congestion-action-profile** コマンドを使用すると、特定のしきい値およびしきい値レベルに対して実行する追加のアクションをオペレータが確立できます。

## 輻輳制御の設定

輻輳制御機能を設定するには、次の手順を実行します。

- ステップ 1** の説明に従って、輻輳制御のしきい値を設定します。[輻輳制御のしきい値の設定 \(300 ページ\)](#)
- ステップ 2** の説明に従って、サービス輻輳ポリシーを設定します。[サービス輻輳ポリシーの設定 \(301 ページ\)](#)
- ステップ 3** の説明に従って、リダイレクトオーバーロードポリシーを有効にします。[輻輳制御リダイレクトオーバーロードポリシーの有効化 \(302 ページ\)](#)
- ステップ 4** の説明に従って、コールまたは非アクティブ時間に基づくサブスクリバの接続解除を設定します。[コール時間または非アクティブ時間に基づくサブスクリバの接続解除 \(303 ページ\)](#)
- ステップ 5** 「設定の確認と保存」の章の説明に従って、設定を保存します。

## 輻輳制御のしきい値の設定

輻輳制御のしきい値を設定するには、CLI のグローバル コンフィギュレーション モードで次の設定例を使用します。

```
configure
  congestion-control threshold max-sessions-per-service-utilization
  percent
  congestion-control threshold tolerance percent
end
```

注：

- さまざまなしきい値パラメータがあります。詳細については、『*Command Line Interface Reference*』の「*Global Configuration Mode Commands*」を参照してください。
- 許容度は、設定されたしきい値未満のパーセンテージであり、条件がどの段階でクリアされるかを決定します。
- さまざまなタイプの輻輳制御しきい値で複数レベルの輻輳しきい値（critical、major、および minor）がサポートされています。しきい値レベルが指定されていない場合、デフォルトは critical です。現在、major と minor のしきい値は、MME でのみサポートされています。lte-policy での congestion-action-profile コマンドは、しきい値を超えたときに実行されるアクションを定義します。詳細については、『*Command Line Interface Reference*』の「*Global Configuration Mode Commands*」、「*LTE Policy Configuration Mode Commands*」、および「*Congestion Action Profile Configuration Mode Commands*」を参照してください。
- 追加のしきい値については、必要に応じてこの設定を繰り返します。

## サービス輻輳ポリシーの設定

輻輳制御ポリシーを作成するには、CLI のグローバル コンフィギュレーション モードで次の設定例を適用します。

```
configure
  congestion-control policy service action { drop | none | redirect
  | reject }
end
```

注：

- PDSN サービスに対してリダイレクトアクションが発生すると、PDSN は、代替 PDSN の IP アドレスとともに 136 の応答コード「unknown PDSN address」を使用して PCF に応答します。
- **redirect** は、PDIF では使用できません。PDIF のデフォルトのアクションは「なし」です。
- HA サービスに対してリダイレクトアクションが発生すると、システムは応答コード 136 「unknown home agent address」を使用して FA に応答します。
- **redirect** は、GGSN サービスと組み合わせて使用することはできません。
- **redirect** は、ローカルモビリティアンカー（LMA）サービスでは使用できません。
- アクションを [reject] に設定すると、応答コードは 130、「insufficient resources」になります。
- GGSN では、応答コードは 199、「no resources available」です。
- SaMOG、MME、では、**redirect** は使用できません。
- MME では、グローバル コンフィギュレーション モードで **lte-policy** の下にある **congestion-action-profile** コマンドを使用して、オプションのメジャーしきい値とマイナーしきい値のアクションプロファイルを作成します。

- MMEでは、サービスをクリティカル、メジャー、またはマイナーとして指定し、ポリシーを設定してそれぞれのしきい値にアクションプロファイルを関連付けることができます。詳細については、『*Command Line Interface Reference*』の「*Global Configuration Mode Commands*」を参照してください。

## MMEでのオーバーロードレポートの設定

過負荷状況がMMEで検出され、**congestion-control policy** コマンドで **report-overload** キーワードが有効になっている場合、システムはその状況を指定されたパーセンテージのeNodeBsに報告し、着信セッションに対して設定されたアクションを実行します。過負荷レポートを使用して輻輳制御ポリシーを作成するには、次の設定例を適用します。

```
configure
  congestion-control policy mme-service action report-overload
  reject-new-sessions enodeb-percentage percentage
end
```

注：

- その他の過負荷アクションには、**permit-emergency-sessions** および **reject-non-emergency-sessions**が含まれます。

## 輻輳制御リダイレクトオーバーロードポリシーの有効化

輻輳制御ポリシーを作成し、サービスにリダイレクトオーバーロードポリシーを設定するには、次の設定例を適用します。

```
configure
  congestion-control
    context context_name
      {service_configuration_mode}
    policy overload redirect address
  end
```

注：

- 任意：輻輳制御ポリシーアクションが **redirect** に設定されている場合は、影響を受けるサービスに対してリダイレクトオーバーロードポリシーを設定する必要があります。
- 複数のサービス コンフィギュレーション モードを設定できます。モードの完全なリストについては、『*Command Line Interface Reference*』を参照してください。
- リダイレクションにはさまざまなオプションを設定できます。詳細については、『*Command Line Interface Reference*』を参照してください。
- 同じコンテキストで設定された追加サービスにオーバーロードポリシーを設定するには、この設定例を繰り返します。

## サービスオーバーロードポリシーの検証

サービスオーバーロードポリシーが適切に設定されていることを確認するには、Execモードで次のコマンドを入力します。

```
[local]host_name# show service_type name service_name
```

このコマンドは、サービス設定全体を表示します。「オーバーロードポリシー」に表示されている情報が正確であることを確認します。

他のコンテキストで追加サービスを設定するには、この設定例を繰り返します。

## 輻輳制御の設定の確認

### MMEの輻輳アクションプロファイルの確認

MMEのマルチレベルの輻輳アクションプロファイルを確認するには、次のExecモードのコマンドを実行します。

```
[local]host_name# show lte-policy congestion-action-profile { name profile_name  
| summary }
```

### コール時間または非アクティブ時間に基づくサブスクリバの接続解除

システム負荷が高いときは、許容レベルのシステムパフォーマンスを維持するために、サブスクリバの切断が必要になる場合があります。しきい値を設定して、コールが接続または非アクティブになっている時間に基づいて、コールを切断するサブスクリバを選択できます。

現在選択されているサブスクリバの過負荷切断を有効にするには、次の設定例を使用します。

```
configure  
  context context_name  
    subscriber name subscriber_name  
      default overload-disconnect threshold inactivity-time dur_thresh  
  
      default overload-disconnect threshold connect-time dur_thresh  
    end
```

サブスクリバの過負荷切断機能を無効にするには、次の設定例を使用します。

```
configure  
  context context_name  
    subscriber subscriber_name  
      no overload-disconnect { [threshold inactivity-time] | [threshold  
connect-time] }  
    end
```

■ コール時間または非アクティブ時間に基づくサブスクリバの接続解除





## 第 22 章

# ルーティング

この章では、拡張または拡張サービスの設定について説明します。製品アドミニストレーションガイドには、システム上での基本サービスの設定例と手順が示されています。以下で説明する手順を使用する前に、サービスモデルに最適な設定例を選択し、そのモデルに必要な要素を設定する必要があります。

この章は、次の項で構成されています。

- [ルーティング ポリシー \(305 ページ\)](#)
- [スタティック ルーティング \(307 ページ\)](#)
- [OSPF ルーティング \(309 ページ\)](#)
- [OSPFv3 ルーティング \(311 ページ\)](#)
- [等コストマルチパス \(ECMP\) \(313 ページ\)](#)
- [BGP-4 ルーティング \(314 ページ\)](#)
- [双方向フォワーディング検出 \(325 ページ\)](#)
- [ルーティング情報の表示 \(335 ページ\)](#)

## ルーティング ポリシー

この項では、ルーティングポリシーを定義するために必要な要素を設定する方法について説明します。ルーティングポリシーは、特定のネットワーク展開要件を満たすために、システムとの間でルートを変更およびリダイレクトします。

ルーティングポリシーを設定するには、次の構成要素を使用します。

- **ルートアクセスリスト**：ルーティングポリシーの基本的な構成要素です。ルートアクセスリストは、IP アドレスの範囲に基づいてルートをフィルタ処理します。
- **IP プレフィックスリスト**：より高度なルーティングポリシーの要素です。IP プレフィックスリストは、IP プレフィックスに基づいてルートをフィルタ処理します。
- **AS パスアクセスリスト**：ボーダーゲートウェイプロトコル (BGP) のルーティングに使用される基本的な構成要素です。これらのリストは、自律システム (AS) のパスをフィルタ処理します。

- **ルートマップ**：ルートマップは、ルーティングプロトコルによるルートを選択またはルートのアドバタイズメントの際、およびルーティングプロトコル間のルートの再配布において、ルートマップの詳細な制御を提供します。このレベルの制御では、IPプレフィックスリスト、ルートアクセスリスト、およびASパスアクセスリストを使用して、IPアドレス、アドレスの範囲、および自律システムパスを指定します。

## IP プレフィックスリストの作成

IP プレフィックスリストを作成するには、次の設定例を使用します。

```
config
  context context_name
    ip prefix-list name list_name { deny | permit } network_address/net_mask
```

注：

- IPプレフィックスリストを [deny]、[permit]、または [match any prefix] に設定します。
- IPv4 ドット付き 10 進数および IPv6 コロン区切り 16 進数のアドレスがサポートされています。
- 「設定の確認と保存」の章の説明に従って、設定を保存します。

## ルートアクセスリストの作成

ルートアクセスリストを作成するには、次の手順を実行します。

```
config
  context context_name
    route-access-list { extended identifier } { deny | permit } [ ip
address ip_address ]
    route-access-list named list_name { deny | permit } { ip_address/mask |
any } [ exact-match ]
  route-access-list
  standard identifier { permit | deny ) { ip_address
wildcard_mask | any |network_address }
```

注：

- コンテキストごとに最大 64 のアクセスリストがサポートされます。
- 各ルートアクセスリストに対して最大 16 のエントリを定義できます。
- 「設定の確認と保存」の章の説明に従って、設定を保存します。

## AS パスアクセスリストの作成

AS パスアクセスリストを作成するには、次の手順を実行します。

```
config
  context context_name
    ip as-path access-list list_name [ { deny | permit } req_expr ]
```

注：

- 「設定の確認と保存」の章の説明に従って、設定を保存します。

## ルートマップの作成

ルートマップを作成するには、次の設定例を使用します。

```
config
  context context_name
    route-map map_name { deny | permit } seq_number
```

注：

- ルートマップを設定するには、ルートマップ コンフィギュレーション モードで **match** コマンドと **set** コマンドを使用します。これらのコマンドの詳細については、『*Command Line Interface Reference*』を参照してください。
- 「設定の確認と保存」の章の説明に従って、設定を保存します。

## 設定例

次に、2つのルートアクセスリストを作成し、それらをルートマップに適用し、そのルートマップを BGP ルータネイバーに使用する設定の例を示します。

```
config
  context ispl
    route-access-list named RACLin1a permit 88.151.1.0/30
    route-access-list named RACLin1b permit 88.151.1.4/30
    route-access-list named RACLany permit any
    route-map RMnet1 deny 100
      match ip address route-access-list RACLin1a
      #exit
    route-map RMnet1 deny 200
      match ip address route-access-list RACLin1b
      #exit
    route-map RMnet1 permit 1000
      match ip address route-access-list RACLany
      #exit
  router bgp 1
    neighbor 152.20.1.99 as-path 101
    neighbor 152.20.1.99 route-map RMnet1 in
```

## スタティック ルーティング

システムでは、コンテキストごとにスタティック ネットワークルートの設定がサポートされています。次のように指定して、ネットワークルートを定義します。

- ルートの IP アドレスとマスク
- ルートが使用する必要がある現在のコンテキスト内のインターフェイスの名前
- ネクスト ホップの IP アドレス



**重要** では、IPv6 プレフィックス長が /12 未満で、/64 と /128 の範囲のスタティックルートはサポートされていません。

## コンテキストへのスタティックルートの追加

コンテキストコンフィギュレーションにスタティックルートを追加するには、現在のコンテキストで設定されているインターフェイスの名前を把握する必要があります。現在のコンテキスト (Exec モード) のインターフェイスを一覧表示するには、**show ip interface** コマンドを使用します。

次の例に示すように、現在のコンテキストで設定されているすべてのインターフェイスの情報が表示されます。

```
[local]host_name# show ip interface
Intf Name: Egress 1
Description:
IP State: Up (Bound to slot/port untagged ifIndex 402718721)
IP Address: 192.168.231.5
Subnet Mask: 255.255.255.0
Bcast Address: 192.168.231.255
MTU: 1500
Resoln Type: ARP          ARP timeout: 3600 secs
L3 monitor LC-port switchover: Disabled
Number of Secondary Addresses: 0
Total interface count: 1
```

出力例に示されているように、各インターフェイスの最初の情報行には、現在のコンテキストのインターフェイス名が一覧表示されます。この例では、*Egress 1* という名前のインターフェイスが 1 つあります。

```
config
  context context_name
    ip route { ip_address [ ip_mask ] | ip_addr_mask_combo } { next-hop
next_hop_address | egress_name [ precedence precedence [ cost cost ]
```

注：

- コンテキストごとに最大 1,200 のスタティックルートを設定できます。「設定の確認と保存」の章の説明に従って、設定を保存します。

## コンテキストからのスタティックルートの削除

コンテキストの設定からスタティックルートを削除するには、次の設定例を使用します。

```
config
  context context_name
    no ip route { ip_address ip_mask | ip_addr_mask_combo } next_hop_address
    egress_name [ precedence precedence ] [ cost cost ]
```

注意

- ・「設定の確認と保存」の章の説明に従って、設定を保存します。

## OSPF ルーティング

この項では、システムでの Open Shortest Path First (OSPF) ルーティングの概要とその実装を示します。また、基本的な OSPF 機能を有効にする方法と、より複雑な設定に使用できるコマンドのリストを示します。

この機能を使用するには、ライセンスキーを購入してインストールする必要があります。ライセンスの詳細については、シスコのアカウント担当者にお問い合わせください。



**重要** リカバリ中にダイナミック ルーティング プロトコルから転送エントリが削除された場合は、動的に学習された転送エントリがシステムタスクのリカバリ時にシステム転送テーブルに誤って残る可能性があります。



**重要** では、IPv6 プレフィックス長が /12 未満で、/64 ~ /128 の範囲の OSPF ルートはサポートされていません。

## OSPF バージョン 2 の概要

OSPF は、内部ゲートウェイプロトコル (IGP) を使用し、IP パケットヘッダー内の接続先 IP アドレスのみに基づく最短パスを最初に使用して IP パケットをルーティングするリンクステートルーティング プロトコルです。OSPF ルーテッド IP パケットは、ネットワークを通過するときに追加のプロトコルヘッダー内にカプセル化されません。

自律システム (AS) またはドメインは、共通のルーティング インフラストラクチャ内のネットワークのグループとして定義されます。

OSPF は、AS 内のトポロジの変化 (ルータインターフェイスの障害など) を迅速に検出し、コンバージェンスの期間後に新しいループフリールートを実算するダイナミック ルーティング プロトコルです。このコンバージェンスの期間は短く、関与するルーティングトラフィックは最小限となります。

リンクステートルーティング プロトコルでは、各ルータは、自律システムのトポロジを記述する、リンクステートデータベースと呼ばれるデータベースを維持します。参加している各ルータには同一のデータベースが備わっています。このデータベースの各エントリは、特定の

ルータのローカル状態（ルータの使用可能なインターフェイスや到達可能なネイバーなど）です。ルータは AS 全体にローカル状態であることをフラッドングにより配信します。

すべてのルータが同じアルゴリズムを並行して実行します。各ルータはそれ自体を AS 内の各接続先へのルートとした最短パスのツリーをリンクステートデータベースから構築します。外部から取得したルーティング情報は、リーフとしてツリーに表示されます。ルートのコストは、1つの無次元メトリックで記述されます。

OSPF では、一連のネットワークをまとめてグループ化できます。このようなグループ化をエリアと呼びます。このエリアのトポロジは AS の残りの部分からは隠されているため、ルーティングトラフィックの大幅な削減が可能になります。また、エリア内のルーティングはエリア自体のトポロジによってのみ決定されるため、不正なルーティングデータからエリアを保護できます。エリアとは、IP サブネット化ネットワークの汎化です。

OSPF では IP サブネットを柔軟に設定でき、OSPF によって配信された各ルートに接続先とマスクを持たせることができます。同じ IP ネットワーク番号の 2 つの異なるサブネットでは、サイズが異なる（つまり、マスクが異なる）場合があります。これは一般に可変長サブネットと呼ばれています。パケットは最適な（最長または最も具体的な）一致にルーティングされます。ホストルートは、マスクが「すべて 1」（0xffffffff）であるサブネットと見なされます。

OSPF トラフィックは、認証されるかまたは非認証となります。また、認証、簡易/クリアテキストのパスワード、または MD5 ベースのパスワードは使用できません。つまり、信頼できるルータのみが AS ルーティングに参加できることを意味します。さまざまな認証スキームを指定でき、実際には、IP サブネットごとに個別の認証スキームを設定できます。

外部から取得したルーティングデータ（BGP などの外部プロトコルから学習したルートなど）は AS 全体にアドバタイズされます。この外部から取得したデータは、OSPF のリンクステートデータとは別に保持されます。

また、各外部ルートはアドバタイズルータによってタグ付けされ、AS の境界にあるルータ間で追加情報を渡すこともできます。

OSPF は、リンクステートアルゴリズムを使用して、すべての既知の宛先までの最短パスを構築および計算します。

## OSPFv2 設定の基本

この項では、基本的な OSPF ルーティングの実装方法を説明します。

### 特定のコンテキストの OSPF ルーティングの有効化

特定のコンテキストに対して OSPF ルーティングを有効にするには、次の設定例を使用します。

```
config
  context context_name
    router ospf
  end
```

注：

- 「設定の確認と保存」の章の説明に従って、設定を保存します。

## 特定のインターフェイスでの OSPF の有効化

OSPF を有効にした後、実行するネットワークを指定します。次のコマンドを使用して、OSPF を有効にします。

```
network network_ip_address/network_mask area { area_id | area_ip_address }
```



---

**重要** システム上の OSPF のデフォルトコストは 10 です。コストを変更するには、『*Command Line Interface Reference*』の「*Ethernet Interface Configuration Mode Commands*」の章に記載されている **ip ospf cost** コマンドを参照してください。

---

注：

- 「設定の確認と保存」の章の説明に従って、設定を保存します。

## OSPF へのルートの再配布（オプション）

ルートを OSPF に再配布するということは、ルートタイプ、メトリック、ルートマップ内のルールなど、指定された基準を満たす別のプロトコルからのすべてのルートが、OSPFv2 プロトコルを使用してすべての OSPF エリアに再配布されることを意味します。この設定は、オプションです。

```
config
  context context_name
    router ospf
      redistribute { connected | static }
    end
```

注：

- 「設定の確認と保存」の章の説明に従って、設定を保存します。

## OSPF 設定パラメータの確認

OSPF ルータの設定を確認するには、次のコマンドを使用し、画面出力で **router OSPF** というセクションを探します。

```
show config context ctxt_name [ verbose ]
```

## OSPFv3 ルーティング

この項では、システムでの Open Shortest Path First バージョン 3（OSPFv3）のルーティングとその実装の概要を示します。また、基本的な OSPFv3 機能を有効にする方法と、より複雑な設定に使用できるコマンドのリストを示します。



重要 では、IPv6 プレフィックス長が /12 未満で、/64 ~ /128 の範囲の OSPFv3 ルートはサポートされていません。

## OSPFv3 の概要

OSPF バージョン 3 の大部分は OSPF バージョン 2 と同じです。OSPFv3 は IPv6 ルーティングプレフィックスと大型の IPv6 アドレスをサポートするように OSPF バージョン 2 が拡張されています。OSPFv3 は、OSPFv3 ルーティングドメイン内の IPv6 ルートを動的に学習し、アドバタイズ（再配布）します。

OSPFv3 では、ルーティングプロセスを明示的に作成する必要はありません。インターフェイス上で OSPFv3 を有効にすると、ルーティングプロセスとそれに関連する設定が作成されます。

## OSPFv3 設定の基本

この項では、基本的な OSPF ルーティングの実装方法を説明します。

### 特定のコンテキストにおける OSPFv3 ルーティングの有効化

特定のコンテキストに対して OSPF ルーティングを有効にするには、次の設定例を使用します。

```
config
  context context_name
    router ospfv3
  end
```

注：

- 「設定の確認と保存」の章の説明に従って、設定を保存します。

### 特定のインターフェイスでの OSPFv3 の有効化

OSPFv3 を有効にした後は、実行するエリアを指定します。次のコマンドを使用して、OSPFv3 を有効にします。

```
area { area_id | area_ip_address } [ default-cost dflt-cost ] [ stub stub-area ] [ virtual-link vl-neighbor-ipv4address ]
```



重要 システム上の OSPFv3 のデフォルトコストは 10 です。コストを変更するには、『*Command Line Interface Reference*』の「*Ethernet Interface Configuration Mode Commands*」の章に記載されている `ipv6 ospf cost` コマンドを参照してください。



注：

- 「設定の確認と保存」の章の説明に従って、設定を保存します。

## OSPFv3 へのルートの再配布（オプション）

ルートを OSPFv3 に再配布するという事は、ルートタイプ、メトリック、ルートマップ内のルールなど、指定された基準を満たす別のプロトコルからのすべてのルートが、OSPFv3 プロトコルを使用してすべての OSPF エリアに再配布されることを意味します。この設定は、オプションです。

```
config
  context context_name
    router ospf3
      redistribute { connected | static }
    end
```

注：

- 「設定の確認と保存」の章の説明に従って、設定を保存します。

## OSPFv3 設定パラメータの確認

OSPF ルータの設定を確認するには、次のコマンドを使用して、画面出力で「**router ipv6 ospf**」というラベルの付いたセクションを探します。

```
show config context ctxt_name [ verbose ]
```

## 等コストマルチパス（ECMP）

システムは、ルーティングプロトコルの ECMP をサポートしています。ECMP は、1 つのルートの負担を軽減するために、コストが同じである複数のルートにトラフィックを分散します。

ECMP は、ほとんどのルーティングプロトコルと組み合わせて使用できます。これは、ルーティングプロトコルはホップごとに決定され、1 つのルータに限定されるためです。複数のパスでトラフィックをロードバランシングすることにより、帯域幅が大幅に増加する可能性があります。

次のコマンドは、ルーティングプロトコルによって送信可能な等コストパスの最大数を設定します。

```
config
  context context_name
    ip routing maximum-paths [ max_num ]
```

注：

- *max\_num* は 1 ～ 32 の整数です。
- 「設定の確認と保存」の章の説明に従って、設定を保存します。

## BGP-4 ルーティング

ボーダーゲートウェイプロトコル4 (BGP-4) ルーティングプロトコルは、コンテキストレベルで実装されている BGP ルータプロセスを介してサポートされています。

ボーダーゲートウェイプロトコル (BGP) は、AS間ルーティングプロトコルです。自律システム (AS) は、内部ゲートウェイプロトコルと AS内のパケットをルーティングするための共通メトリックを使用する 1つの技術的な管理下にあるルータのセットです。ルータのセットは、外部ゲートウェイプロトコルを使用して、他の自律システムにパケットをルーティングします。

BGP は TCP を介して実行されます。これにより、BGP プロトコルによる明示的な更新のフラグメント化、再送信、確認応答、およびシーケンシング情報の実装が不要になります。BGP 独自の認証メカニズムに加えて、TCP で使用される認証方式が使用される場合があります。

BGP ルータは、他の BGP ルータとネットワーク到達可能性情報を交換します。この情報によって、ルートがフィルタ処理され、AS レベルのポリシー決定が適用される AS 接続の実像がつかめます。

BGP-4には、クラスレスドメイン間ルーティングが備わっています。これには、IPプレフィックスをアダプタイズするためのサポートが含まれており、BGP内のネットワーククラスという概念を排除します。また、BGP 4 では、AS パスの集約を含むルートの集約も可能です。



---

**重要** では、IPv6 プレフィックス長が /12 未満と /64 ~ /128 の範囲の BGP ルートはサポートされていません。

---

## BGP サポートの概要

モバイルデバイスは、ホームエージェント (HA) を通じてインターネットと通信します。HA は、設定されたアドレスプールから IP アドレスをモバイルノードに割り当てます。これらのアドレスは、ダイナミックルーティングを保証するために、IPルーティングプロトコルを使用してインターネットルータにもアダプタイズされます。BGP-4プロトコルは、シャージ間セッションリカバリ (ICSR) をサポートするためのルーティングを備えた HA とインターネットルータ間のモニタリングメカニズムとして使用されます。(詳細については、「シャージ間セッションリカバリ」を参照してください)。

BGP-4プロトコルのサポートの目的は、ルーティング要件を満たし、インターネットルータとの通信をモニターすることです。BGP-4は、サブスクリバサービスの中断を防ぐために、アクティブからスタンバイへのスイッチオーバーをトリガーする場合があります。

サポートされる BGP-4 機能は次のとおりです。

- 外部ボーダー ゲートウェイ プロトコル (EBGP) マルチホップ
- インバウンドおよびアウトバウンドルートのルートフィルタリング

- ルート再配布とルートマップ
- ルートマップでの BGP コミュニティおよび BGP 拡張コミュニティのサポート
- IPv4 および IPv6 (IBGP ピア) のローカルプリファレンス

IP プールルートとループバックルートは、次の方法で BGP ドメインでアドバタイズされます。

- BGP コンフィギュレーションモードの **redistribution** コマンドを使用すると、接続されているすべてのルートまたはその一部が BGP ドメインに再配布されます。(IP プールとループバックルートは、接続されたルートとして IP ルーティングテーブルに存在します)。**network routemap** コマンドを使用すると、多くの BGP 属性を柔軟に変更できます。
- BGP コンフィギュレーションモードの **network** コマンドを使用すると、接続されたルートは BGP ドメインにアドバタイズされるように明示的に設定されます。**network routemap** コマンドを使用すると、多くの BGP 属性を柔軟に変更できます。これらのコマンドの詳細については、『*Command Line Interface Reference*』の「*BGP Configuration Mode Commands*」の章を参照してください。



**重要** 処理カードの障害、移行、クラッシュ、または処理カードの削除のために BGP タスクが再起動した場合は、すべてのピアリングセッションとルート情報が失われます。

## BGP の設定

この項では、システムでの基本的な BGP ルーティングサポートを設定し、有効にする方法について説明します。

**config**

```
context context_name
  router bgp AS_number
    neighbor ip_address remote-as AS_num
```

注：

- コンテキストごとに最大 64 の BGP ピアがサポートされます。
- 「設定の確認と保存」の章の説明に従って、設定を保存します。

## BGP へのルートの再配布 (オプション)

ルートを BGP に再配布するということは、ルートタイプ、ルートマップ内のルールなど、指定された基準を満たす別のプロトコルからのすべてのルートが、BGP プロトコルを使用してすべての BGP エリアに再配布されることを意味します。この設定は、オプションです。

**config**

```
context context_name
  router bgp as_number
```

```

redistribute bgp { bgp | connected | static } [ metric metric_value
] [ metric-type { 1 | 2 } ] [ route-map route_map_name ]

```

注：

- 再配布オプションは、connected、ospf、rip、または static です。redistribute コマンドの詳細については、『*Command Line Interface Reference*』の「*Border Gateway Protocol Configuration Mode Commands*」の章を参照してください。
- コンテキストごとに最大 64 のルートマップがサポートされます。
- 「設定の確認と保存」の章の説明に従って、設定を保存します。

## BGP コミュニティと BGP 拡張コミュニティ

BGP コミュニティまたは拡張コミュニティ（ルートターゲット）に基づくルートフィルタリングは、CLI ルートマップのコンフィギュレーションモードのコマンドを介して設定します。

### BGP コミュニティ

#### BGP コミュニティの設定

BGP コミュニティは、共通するいくつかの属性を共有する接続先のグループです。各宛先は複数のコミュニティに属します。自律システム管理者は、接続先が属するコミュニティを定義します。

BGP コミュニティは、コンテキスト コンフィギュレーションモードのコマンドを使用して設定します。

```

config
context context_name
  ip community-list { named named_list | standard identifier } { deny
| permit } { internet | local-AS | no-advertise | no-export | value
AS-community_number AS-community_number AS-community_number ... }
  { internet | local-AS | no-advertise | no-export | value
AS-community_number AS-community_number AS-community_number ... }
  { internet | local-AS | no-advertise | no-export | value
AS-community_number AS-community_number AS-community_number ... }

```

次の BGP コミュニティの接続先を許可または拒否できます。

- **internet**：このルートをインターネット コミュニティと、そのコミュニティに属するあらゆるルータにアドバタイズします。
- **local-AS**：パケットがローカルの自律システム（AS）外部へ送信されないようにするために、コンフェデレーション シナリオで使用します。
- **no-advertise**：どの BGP ピア（内部または外部）にもこのルートをアドバタイズしません。
- **no-export**：外部 BGP（eBGP）ピアへアドバタイズしません。このルートは AS 内に留められます。
- **value community\_number**：AS:NN 形式のコミュニティ文字列を指定します。ここで、AS は 2 バイトの AS コミュニティ 16 進数、NN は 2 バイトの 16 進数です（1～11 文字）。

各コミュニティには、複数の接続先と AS コミュニティ番号を入力できます。詳細については、『*Command Line Interface Reference*』を参照してください。

複数のコミュニティリストエントリをコミュニティリストに付加するには、さまざまなコミュニティストリングに対して複数の `permit` 句または `deny` 句を追加します。最大 64 のコミュニティリストをコンテキストで設定できます。

## コミュニティ属性の設定

BGP コミュニティ属性は、ルートマップ内の `set community` コマンドを介して設定します。

```

config
  context context_name
    route-map map_name { deny | permit } sequence_number
      set community [additive]{ internet | local-AS | no-advertise
| no-export | none | value AS-community_number AS-community_number
AS-community_number ...}
      { internet | local-AS | no-advertise | no-export | none | value
AS-community_number AS-community_number AS-community_number ... }
      { internet | local-AS | no-advertise | no-export | none | value
AS-community_number AS-community_number AS-community_number ... }

```

`additive` オプションを指定すると、複数の接続先および AS コミュニティ番号を入力できます。詳細については、『*Command Line Interface Reference*』を参照してください。

## BGP コミュニティを介したフィルタリング

BGP コミュニティに基づいてルートをフィルタ処理するには、ルートマップで `match` 句を設定します。コマンドシーケンスは次のとおりです。

```

config
  context context_name
    route-map map_name { deny | permit } sequence_number
      match community { named named_list | standard identifier }

```

## BGP 拡張コミュニティ

### BGP 拡張コミュニティの設定（ルートターゲット）

BGP 拡張コミュニティは、ルートターゲットを定義します。MPLS VPN は、ルートターゲット（RT）と呼ばれる 64 ビットの拡張コミュニティ属性を使用します。RT は、適切な情報テーブルへの到達可能性情報の配信を可能にします。

BGP 拡張コミュニティは、コンテキスト コンフィギュレーション モードのコマンドを使用して設定します。

```

config
  context context_name
    ip extcommunity-list { named named_list | standard identifier } {
deny | permit } rt rt_number rt_number rt_number ...

```

`rt_number` は AS:NN 形式の文字列としてルートターゲットを指定します。ここで、AS は 2 バイトの AS コミュニティ 16 進数、NN は 2 バイトの 16 進数です (1 ~ 11 文字)。IP `extcommunity` リストには複数のルート番号を追加できます。

さまざまな拡張コミュニティ文字列に複数の `permit` 句または `deny` 句を追加することで、拡張コミュニティリストに複数の拡張コミュニティリストエントリを付加できます。最大 64 個の拡張コミュニティリストをコンテキスト内に設定できます。

## 拡張コミュニティ属性の設定

ルートマップ内で `set extcommunity` コマンドを使用して BGP 拡張コミュニティ属性を設定します。

### config

```
context context_name
  route-map map_name { deny | permit } sequence_number
  set extcommunity rt rt_number rt_number rt_number ...
```

`rt_number` は AS:NN 形式の文字列としてルートターゲットを指定します。ここで、AS は 2 バイトの AS コミュニティ 16 進数、NN は 2 バイトの 16 進数です (1 ~ 11 文字)。IP `extcommunity` リストには複数のルート番号を追加できます。

## BGP 拡張コミュニティを介したフィルタリング

BGP 拡張コミュニティ (ルートターゲット) に基づいてルートをフィルタ処理するには、ルートマップで `match` 句を設定します。コマンドシーケンスは次のとおりです。

### config

```
context context_name
  route-map map_name { deny | permit }
  [no] match extcommunity { named named_list | standard identifier
}
```

## BGP ローカルプリファレンス

BGP ローカルプリファレンス属性は、IBGP ピアにのみ BGP スピーカーによって送信されません。次のコマンドシーケンスを使用してルートマップ内に設定されます。

### config

```
context context_name
  route-map map_name { deny | permit }
  set local-preference pref_number
```

ルートマップのローカルプリファレンスに対応する `match` 句はありません。これは、ローカルプリファレンスがルート選択アルゴリズムで直接使用されるためです。

## ICSR および SRP グループ

BGP は、サービス冗長性プロトコル (SRP) を介してリンクされたシャーシ間セッションリカバリ (ICSR) の設定で採用されています。デフォルトでは、ICSR フェールオーバーは、コンテキスト内のすべての BGP ピアがダウンしたときにトリガーされます。

必要に応じて、コンテキスト内に SRP ピアグループを設定できます。ICSR フェールオーバーは、グループ内のすべてのピアで障害が発生した場合に発生します。このオプションは、IPv4 と IPv6 のピアの組み合わせが複数ペアの VLAN に分散され、ピアグループのすべてのメンバーによって IPv4 または IPv6 の接続が失われる展開で役立ちます。

詳細については、このガイドの「シャーシ間セッションリカバリ」、および『*Command Line Interface Reference*』の「*Service Redundancy Protocol Configuration Mode Commands*」の章にある **monitor bgp**、**monitor diameter** および **monitor authentication-probe** コマンドの説明を参照してください。

## スタンバイ ICSR シャーシからの BGP ルートのアドバタイジング

SRP コンフィギュレーションモードコマンドは、スタンバイ状態で ICSR シャーシからの BGP ルートのアドバタイジングを有効にします。このコマンドとそのキーワードを使用すると、オペレータは、光トランスポート ネットワーク ジェネレーション ネクスト (OTNGN) で BGP プレフィックス独立コンバージェンス (PIC) を展開することで、より高速なネットワーク コンバージェンスを利用することができます。

BGP PIC は、アグレッシブな ICSR 障害検出タイマーを安全に設定できるネットワーク コンバージェンスを向上させることを目的としています。

```
configure
  context context_name
    service-redundancy-protocol
      advertise-routes-in-standby-state [ hold-off-time hold-off-time ]
      [ reset-bfd-nbrs bfd-down-time ]
    end
```

注：

- **hold-off-time hold-off-time** は、タイマーが期限切れになるまで、BGP ルートのアドバタイジングを遅延させます。1 ~ 300 の整数として *hold-off-time* を秒単位で指定します。
- BFD をリセットした後、**reset-bfd-nbrs bfd-down-time** は、BFD セッションを設定されたミリ秒数の間停止して、ネットワーク コンバージェンスを向上させます。*bfd-down-time* を 50 ~ 120000 の整数として指定します。

## ICSR の設定可能な BGP ルートのアドバタイズメント間隔

デフォルトでは、MinRtAdvInterval が各ピアに設定されます。その値は iBGP ピアの場合は 5 秒、eBGP ピアの場合は 30 秒です。オペレータは、**neighbor identifier advertisement-interval** コマンドを使用して、デフォルトの間隔をグローバルに変更できます。

BGP advertisement-interval は、アドレスファミリーごとに個別に設定することもできます。設定されている場合、この値は、そのアドレスファミリーにおけるピアのデフォルトの advertisement-interval のみをオーバーライドします。BGP は、AFI/SAFI に設定された advertisement-interval に基づいて、AFI/SAFI ごとにルート更新メッセージを送信します。AFI/SAFI advertisement-interval が設定されていない場合は、ピアベースのデフォルトの advertisement-interval が使用されます。

ICSR 設定では、この機能を使用して、ルートアドバタイズメントを高速化し、ネットワークコンバージェンス時間を短縮することができます。

**timers bgp icshr-aggr-advertisement-interval** コマンドは、BGP アドレスファミリー (VPNv4/VPNv6) コンフィギュレーションモードと BGP アドレスファミリー (VRF) コンフィギュレーションモードの両方で使用できます。

#### configure

```
context context_name
router bgp as_number
address-family { ipv4 | ipv6 | vpnv4 | vpnv6 }
timers bgp icshr-aggr-advertisement-interval seconds
```

注：

- *seconds* : 0 ~ 30 の整数で秒数を設定します。デフォルト : 0。

## BGP CLI 設定コマンド

次の表に、さまざまな BGP パラメータの設定をサポートする BGP コンフィギュレーションモードの CLI コマンドを示します。詳細については、『*Command Line Interface Reference*』の「*BGP Configuration Mode Commands*」の章を参照してください。

#### configure

```
context context_name
router bgp as_number
```

表 22: BGP コンフィギュレーションモードの CLI コマンド

bgp コマンド	説明
<b>accept-zero-as-rd</b>	管理サブフィールドを持つルート識別子 (RD) 値 (AS 番号 0) がある VPN プレフィックスを受け入れるように設定します。
<b>address-family { ipv4   ipv6 }</b>	IPv4 または IPv6 アドレスファミリー設定モードを開始します。
<b>address-family { vpnv4   vpnv6 }</b>	VPNv4 または VPNv6 アドレスファミリー設定モードを開始します。
<b>bgp graceful-restart { restart-time rest_time   stalepath-time stale_time   update-delay delay</b>	グレースフルリスタートに関する BGP 固有のパラメータを定義します。
<b>description text</b>	この設定の説明テキストを入力できます。
<b>distance { admin distance prefix prefix_addr [ route-access-list list_name ]   bgp external ebgp_dist internal ibgp_dist local local_dist }</b>	ルートのアドミニストレーティブディスタンスを定義します。アドミニストレーティブディスタンスは、特定のルートまたはタイプルートのデフォルトの優先順位です。



bgp コマンド	説明
<b>enforce-first-as</b>	Exterior Border Gateway Protocol (eBGP) ルートの最初の AS を適用します。
<b>ip vrf</b> <i>vrf_name</i>	事前に設定された IP VRF コンテキストインスタンスを BGP ASN に追加し、BGP 属性と関連パラメータを VRF に設定します。
<b>maximum-paths</b> { <b>ebgp</b> <i>max_num</i>   <b>ibgp</b> <i>max_num</i> }	複数のパスを介したパケットの転送を有効にし、イバー間の外部 BGP (eBGP) パスまたは内部 BGP (iBGP) パスの最大数を指定します。
<b>neighbor</b> <i>ip_address</i> { <b>activate</b>   <b>advertisement-interval</b> <i>adv_time</i>   <b>capability graceful-restart</b>   <b>default-originate</b> [ <b>route-map</b> <i>map_name</i> ]   <b>distribute-list</b> <i>dist_list</i> { <b>in</b>   <b>out</b> }   <b>ebgp-multihop</b> [ <b>max-hop</b> <i>number</i> ]   <b>encrypted password</b> <i>encrypted_password</i>   <b>fall-over bfd</b> [ <b>multihop</b> ]   <b>filter-list</b> <i>filt_list</i> { <b>in</b>   <b>out</b> }   <b>max-prefix</b> <i>max_num</i> [ <b>threshold</b> <i>thresh_percent</i> ] [ <b>warning-only</b> ]   <b>next-hop-self</b>   <b>password</b> <i>password</i>   <b>remoteas</b> <i>AS_num</i>   <b>remove-private-AS</b>   <b>restart-time</b> <i>rest_time</i>   <b>route-map</b> <i>map_name</i> { <b>in</b>   <b>out</b> }   <b>send-community</b> { <b>both</b>   <b>extended</b>   <b>standard</b> }   <b>shutdown</b>   <b>srp-activated-soft-clear</b>   <b>timers</b> { [ <b>connect-interval</b> <i>conn_time</i> ]   [ <b>keepalive-interval</b> <i>keep_time</i> <b>holdtimeinterval</b> <i>hold_time</i> ] }   <b>update-source</b> <i>ip_address</i>   <b>weight</b> <i>value</i> }	<p>非ブロードキャスト ネットワークに相互接続する BGP ルータを設定します。他のパラメータを設定する前に、ネイバーに対してリモート AS 番号を指定する必要があります。ご注意ください。</p> <p>注: <b>advertisement-interval</b> は、そのアドレスファミリーに有効になるように、アドレスファミリーに対して明示的に設定する必要があります。デフォルトでは、IPv4 アドレスファミリーにのみ適用されます。<b>address-family</b> コマンドを使用してアドレスファミリーを指定します。その後で、アドレスファミリー コンフィギュレーションモードでネイバーアドバタイズメント間隔を設定できます。</p>
<b>network</b> <i>ip_address/mask</i> [ <b>route-map</b> <i>map_name</i> ]	BGP を介してアナウンスするネットワークを指定します。
<b>redistribute</b> { <b>connected</b>   <b>ospf</b>   <b>rip</b>   <b>static</b> } [ <b>route-map</b> <i>map_name</i> ]	BGP を介して別のプロトコルから BGP にルート再配布します。
<b>router-id</b> <i>ip_address</i>	設定されたルータ ID を上書きし、BGP ピアをリセットします。
<b>scan-time</b> <i>time</i>	BGP バックグラウンドスキャナの間隔を秒単位で設定します。BGP は、インストールされているルータのネクストホップをモニターして、ネクストホップの到達可能性を確認し、BGP ベストパスループを選択、インストール、および検証します。
<b>timers</b> <b>bgp</b> <b>keepalive-interval</b> <i>interval</i> <b>holdtime-interval</b> <i>time</i> [ <b>min-peer-holdtimeinterval</b> <i>time</i> ]	BGP ルーティングタイマーを設定します。

## BGP 設定パラメータの確認

BGP ルータの設定を確認するには、次のコマンドを使用して、画面出力で「**router BGP**」というラベルの付いたセクションを探します。

```
show config context ctxt_name [ verbose ]
```

## BGP ピアの制限

### 機能の概要と変更履歴

#### 要約データ

該当製品または機能エリア	すべて
該当プラットフォーム	VPC - DI
機能のデフォルト	無効：設定が必要
このリリースでの関連する変更点	N/A
関連資料	<ul style="list-style-type: none"> <li>• <i>Command Line Interface Reference</i></li> <li>• <i>Statistics and Counters Reference</i></li> <li>• <i>VPC-DI システム管理ガイド</i></li> </ul>

#### マニュアルの変更履歴

改訂の詳細	リリース
最初の導入。	21.8

### 機能説明

CUPS アーキテクチャを備えたでは、パケット処理カード（逆多重化 SF カードを含むセッション機能（SF））間での BGP ピアリングに柔軟に対応できます。

SDN の「Contrail」モデルに基づく展開手順では、各パケット処理カードにはコンピューティングノード内に vRouter があります。このモデルでは、現在の柔軟な BGP ピアリングスキームを使用して、BGP の設定をこれらの vRouter それぞれに実装する必要があります。そのため、ネットワーク内に多数の SF カードがある場合は、サービスプロバイダーにとっての課題となります。設定が必要な回線数は拡張性の問題をもたらします。

この問題を解決するため、BGP ピア制限機能が導入されました。この機能は BGP ピアリングを VPC-DI アーキテクチャ内の 2 枚の SF カードのみに制限します。この機能では、ルーティングテーブルには、その 2 枚の SF カードに対応するルート 2 つのみが存在し、3 番目のルートは「ブラックホール」または「ヌル」ルートとなります。新しいルートが最長プレフィックスに一致するルートであることを確認するために、ホストアドレスのみ（/32 ビットマスク）のプロビジョニングのみが適用されます。これにより、設定の量とルーティングテーブルのサイズが大幅に削減されます。

## 機能の仕組み

この機能は、**ip route kernel** コマンドを使用して実装されます。設定すると、BGP ピアリングは特別なルートを持つ SF カード 2 枚にのみ制限されます。

**blackhole** キーワードを設定すると、カーネルルーティングエンジンが有効になり、ノードから発信されるパケットをブロックまたはドロップできます。これは、どのインターフェイスにも限定されず、デフォルトではワイルドカードインターフェイスになります。

BGP ピア制限機能の設定の詳細については、「BGP ピア制限の設定」の項を参照してください。

### 制限事項

- この機能のサポートは、コンテキストレベルのみに制限されています。
- VRF レベルではサポートされていません。
- この機能は、IPv4 でのみサポートされています。

## BGP ピア制限の設定

次の項では、この機能を有効または無効にするための設定コマンドについて説明します。

### パケット処理カードのルート設定

次の CLI コマンドを使用して、コンテキスト コンフィギュレーションで定義されている任意の 2 つのパケット処理インターフェイス (SF カード) に特別な (スタティック) ルートを追加します。

```
configure
  context context_name
    [ no ] ip route kernel ip_address/ip_address_mask_combo
  egress_intrfc_name cost number
  end
```

注：

- **no**：追加されたルートを削除します。
- **kernel**：カーネルのルーティングテーブルのオプションでスタティックルートを許可します。
- **ip\_address/ip\_address\_mask\_combo**：ルートが適用される IP アドレスを示すために、結合された IP アドレスのサブネットマスクビットを指定します。**ip\_address\_mask\_combo** は、CIDR 表記を使用して指定する必要があります。これは、IP アドレスが IPv4 ドット付き 10 進表記を使用して指定されたもので、マスクビットは数値 (サブネットマスクのビット数) です。
- **egress\_intrfc\_name**：1 ~ 79 文字の英数字の文字列として、既存の出力インターフェイスの名前を指定します。

- **costnumber** : 次のゲートウェイへのホップ数を定義します。コストは 0 ~ 255 の整数にする必要があります。255が最も高価です。デフォルトは 0 です。
- この機能は、デフォルトでディセーブルになっています。

## ブラックホールルートの設定

ノードから発信されるパケットをブロックまたはドロップするには、次の CLI コマンドを使用します。

```
configure
  context context_name
    [ no ] ip route kernel ip_address/ip_address_mask_combo
    egress_intrfc_name cost number blackhole
  end
```

注 :

- **no** : 追加されたルートを削除します。
- **kernel** : カーネルのルーティングテーブルのオプションでスタティックルートを許可します。
- **ip\_address/ip\_address\_mask\_combo** : ルートが適用される IP アドレスを示すために、結合された IP アドレスのサブネットマスクビットを指定します。 *ip\_address\_mask\_combo* は、CIDR 表記を使用して指定する必要があります。これは、IP アドレスが IPv4 ドット付き 10 進表記を使用して指定されたもので、マスクビットは数値 (サブネットマスクのビット数) です。
- **egress\_intrfc\_name** : 1 ~ 79 文字の英数字の文字列として、既存の出力インターフェイスの名前を指定します。デフォルトは「\*」 (ワイルドカードインターフェイス) です。
- **costnumber** : 次のゲートウェイへのホップ数を定義します。コストは 0 ~ 255 の整数にする必要があります。255が最も高価です。デフォルトは 0 です。
- **blackhole** : パケットをブロックまたはドロップするため、カーネルにインストールするブラックホールルートを定義します。
- この機能は、デフォルトでディセーブルになっています。

## モニターリングおよびトラブルシューティング

この項では、機能のモニターリングとトラブルシューティングのサポートに使用できる CLI コマンドに関する情報を提供します。

### コマンドや出力の表示

この項では、この機能のサポートにおける **show** コマンドやその出力に関する情報を提供します。

## show ip route

この show command CLI は、任意の2つのパケット処理インターフェイス (SF カード) にスタティックルートが追加されると、次の新しいフィールドの値を挿入します。

カーネル専用

## 双方向フォワーディング検出

Bidirectional Forwarding Detection (BFD) は、リンクによって接続されている2つの転送エンジン間の障害を検出するために使用されるネットワークプロトコルです。BFDは、特定のリンクを介した2つのエンドポイント間でセッションを確立します。2つのシステム間に複数のリンクが存在する場合、それぞれをモニターするために複数の BFD セッションが確立される場合があります。セッションは3ウェイハンドシェイクによって確立され、同じ方法で切断されます。セッションで認証が有効になっている場合があります。簡易パスワード、MD5、または SHA1 認証を選択できます。



**重要** では、IPv6 プレフィックス長が /12 未満で、/64 と /128 の範囲の BFD ルートはサポートされていません。

## BFD サポートの概要

BFDには検出メカニズムがありません。エンドポイント間にセッションを明示的に設定する必要があります。基盤となるさまざまな転送メカニズムやレイヤの多くで BFD が使用されている可能性があり、また、それらすべては独立して動作しています。そのため、使用する転送が何であっても、BFD をカプセル化する必要があります。

OSPF や IS-IS などの何らかの形式の隣接セットアップをサポートするプロトコルは、BFD セッションの起動ストラップにも使用できます。これらのプロトコルは、プロトコル独自のキープアライブメカニズムを使用して通常は可能になるよりも迅速に障害が発生しているリンクの通知を受信するために BFD を使用できます。

非同期モードでは、エンドポイントの両方が定期的の Hello パケットを相互に送信しています。これらのパケットを複数回受信しない場合は、セッションがダウンしていると見なされます。

エコーがアクティブになると、エコーパケットのストリームがもう一方のエンドポイントに送信され、その後にそれらが送信元に転送されます。エコーは、bfd protocol コマンドを使用してグローバルに有効にすることも、インターフェイスごとに個別に有効または無効にすることもできます。この機能は、リモートシステム上の転送パスのテストに使用されます。

システムは、スタティックルーティング、または BGP ルーティングを介して、オプションのエコー機能を使用した非同期モードで BFD をサポートしています。



**重要** BFDを機能させるには、ASR5500ではいずれかのパケット処理カードをデマルチプレクサカードとして設定する必要があります。詳細については、「システム設定」の章の「逆多重化カードの設定」の項を参照してください。

## BFD の設定

この項では、システムでの基本的な BFD ルーティングプロトコルのサポートを設定し、有効にする方法について説明します。

BFD プロトコルの設定に影響する要因はいくつかあります。

- [BFD コンテキストの設定 \(326 ページ\)](#)
- [スタティックルートの IPv4 BFD の設定 \(326 ページ\)](#)
- [スタティックルートの IPv6 BFD の設定 \(327 ページ\)](#)
- [シングルホップ用の BFD の設定 \(327 ページ\)](#)
- [マルチホップ BFD の設定 \(328 ページ\)](#)
- [BFD の拡張性 \(328 ページ\)](#)
- [BGP ネイバーとコンテキストの関連付け \(328 ページ\)](#)
- [OSPF ネイバーとコンテキストの関連付け \(329 ページ\)](#)
- [BFD ネイバーグループと BFD プロトコルの関連付け \(329 ページ\)](#)
- [OSPF インターフェイスでの BFD の有効化 \(329 ページ\)](#)
- [ICSR の BFD 接続のモニタリング \(329 ページ\)](#)

### BFD コンテキストの設定

```
config
  context context_name
    bfd-protocol
    [ bfd echo ]
  exit
```

注：

- エコー機能は、このコンテキストのすべてのインターフェイスに対して必要に応じて有効にすることができます。
- コンテキストごとに 16 BFD セッション、およびシャーシごとに 64 BFD セッション。

### スタティックルートの IPv4 BFD の設定

インターフェイス上で BFD を有効にします。

```
config
  context bfd_context_name
    interface if_name
      ip address ipv4_address ipv4_mask
```

```
bfd interval interval_value min_rx rx_value multiplier multiplier_value
[ bfd echo ]
exit
```

BFD スタティックルートを設定します。

```
ip route static bfd if_name ipv4_gw_address
```

スタティックルートを追加します。

```
ip route ipv4_address ipv4_mask
ip route ipv4_address ipv4_mask
```

## スタティックルートの IPv6 BFD の設定

インターフェイスでの BFD を有効にします。

```
config
context bfd_context_name
interface if_name
  ipv6 address ipv6_address ipv6_mask
  bfd interval interval_value min_rx rx_value multiplier multiplier_value
  [ bfd echo ]
exit
```

BFD スタティックルートを設定します。

```
ipv6 route static bfd if_name ipv6_gw_address
```

スタティックルートを追加します。

```
ipv6 route ipv6_address ipv6_mask
ipv6 route ipv6_address ipv6_mask
```



---

**重要** ASR 5500 では、IPv6 プレフィックス長が /12 未満で、/64 ~ /128 の範囲のスタティックルートはサポートされていません。

---

## シングルホップ用の BFD の設定

インターフェイス上で BFD を有効にします。

```
config
context bfd_context_name
interface if_name
  ip address ipv4_address ipv4_mask
  ipv6 address ipv6_address ipv6_mask
  bfd interval interval_value min_rx rx_value multiplier multiplier_value
  [ bfd echo ]
exit
```

BGP ネイバー上で BFD を有効にします。詳細については、[BGP ネイバーとコンテキストの関連付け \(328 ページ\)](#) を参照してください。

OSPF ネイバー上で BFD を有効にします。詳細については、[OSPF ネイバーとコンテキストの関連付け \(329 ページ\)](#) を参照してください。



**重要** ASR 5500 では、IPv6 プレフィックス長が /12 未満で、/64 ~ /128 の範囲のルートはサポートされていません。

## マルチホップ BFD の設定

インターフェイス上で BFD を有効にします。

```
config
context bfd_context_name
interface if_name
ip address ipv4_address ipv4_mask
ipv6 address ipv6_address ipv6_mask
bfd interval interval_value min_rx rx_value multiplier multiplier_value
[ bfd echo ]
exit
```

マルチホップ BFD セッションを設定します。

```
bfd-protocol
bfd multihop peer destination-address interval interval-value multiplier
multiplier-value
```

BGP ネイバー上で BFD を有効にします。詳細については、[BGP ネイバーとコンテキストの関連付け \(328 ページ\)](#) を参照してください。

## BFD の拡張性

上記のいずれかの方法を使用してアクティブな BFD セッションを設定し、アクティブインターフェイスの設定時に同じ BFD ネイバーを使用します。詳細については、[BFD ネイバーグループと BFD プロトコルの関連付け \(329 ページ\)](#) を参照してください。

```
bfd-protocol
bfd nbr-group-name grp_name active-if-name if_name nexthop_address
```

1 つ以上のパッシブインターフェイスに同じ BFD の結果を適用します。

```
bfd nbr-group-name grp_name passive-if-name if_name nexthop_address
bfd nbr-group-name grp_name passive-if-name if_name nexthop_address
```

## BGP ネイバーとコンテキストの関連付け

```
config
context context_name
router bgp AS_number
neighbor neighbor_ip-address remote-as rem_AS_number
neighbor neighbor_ip-address ebgp-multihop max-hop max_hops
neighbor neighbor_ip-address update-source update_src_ip_address
neighbor neighbor_ip-address failover bfd [ multihop ]
```



注：

- ネイバーを追加するには、このシーケンスを繰り返します。

## OSPF ネイバーとコンテキストの関連付け

```
config
  context context_name
    router ospf
      neighbor neighbor_ip-address
```

注：

- ネイバーを追加するには、このシーケンスを繰り返します。

## BFD ネイバーグループと BFD プロトコルの関連付け

```
config
  context context_name
    bfd-protocol
      bfd nbr-group-name grp_name active-if-name if_name nexthop_address
      bfd nbr-group-name grp_name passive-if-name if_name nexthop_address
```

## OSPF インターフェイスでの BFD の有効化

すべての OSPF インターフェイス

```
config
  context context_name
    router ospf
      bfd-all-interfaces
```

特定の OSPF インターフェイス

```
config
  context context_name
    interface interface_name
      broadcast
      ip ospf bfd
```

## ICSR の BFD 接続のモニタリング

ICSR 設定では、次のコマンドシーケンスは、指定されたコンテキストのプライマリシャーシと BFD ネイバーとの間の接続のモニタリングを開始します。接続が切断されると、スタンバイシャーシがアクティブになります。

```
config
  context context_name
    service-redundancy-protocol
      monitor bfd context context_name { ipv4_address | ipv6_address } {
chassis-to-chassis | chassis-to-router }
```

注：

- `ipv4_address / ipv6_address` は、モニターする BFD ネイバーの IP アドレスを定義します。これは、IPv4 ドット付き 10 進表記または IPv6 コロン区切り 16 進表記を使用して入力します。
- `chassis-to-chassis` は、非 SRP リンク上のプライマリシャーシとバックアップシャーシの間で BFD を実行できるようにします。
- `chassis-to-router` は、シャーシとルータの間で BFD を実行できるようにします。

## 設定の保存

「設定の確認と保存」の章の説明に従って、設定を保存します。

## ICSR のシャーシ間 BFD のモニタリング

オペレータは、ICSR スイッチオーバー時にルートをより迅速にアドバタイズするように BFD を設定できます。このソリューションは、スタンバイ ICSR シャーシからの BGP ルートのアドバタイズを可能にする機能を補完します。VoLTE の実装時に ICSR 設定でより積極的な障害検出と回復をサポートすることを全体的な目標としています。

ICSR 設定でシャーシ間 BFD モニタリングを行うには、次の機能を設定する必要があります。

- [プライマリシャーシ BFD モニタリングの有効化 \(330 ページ\)](#)。
- [ICSR Dead 間隔を無視する BFD の設定 \(330 ページ\)](#)。
- [ICSR スイッチオーバー ガードタイマー の設定 \(331 ページ\)](#)。
- [BFD マルチホップ フォールオーバーの有効化 \(331 ページ\)](#)。
- [スタンバイ ICSR シャーシからの BGP ルートのアドバタイジングの有効化 \(333 ページ\)](#)。

## プライマリシャーシ BFD モニタリングの有効化

プライマリシャーシと指定された BFD ネイバー間の接続をモニタリングできるようにする必要があります。接続が切断されると、スタンバイシャーシがアクティブになります。詳細については、[ICSR の BFD 接続のモニタリング \(329 ページ\)](#) を参照してください。

## ICSR Dead 間隔を無視する BFD の設定

SRP コンフィギュレーションモード `bfd-mon-ignore-dead-interval` コマンドを実行すると、スタンバイ ICSR シャーシは dead 間隔を無視し、すべての BFD シャーシ間モニターが失敗するまでスタンバイ状態のままになります。

この機能は、BFD シャーシ間モニタリングとの関連付けで有効にして、よりアグレッシブな ICSR 障害検出時間をサポートします。

**configure**

```
context context_name
  service-redundancy-protocol variable
    bfd-mon-ignore-dead-interval
  end
```

## ICSR スイッチオーバー ガードタイマー の設定

SRP コンフィギュレーションモードの **guard timer** コマンドは、SRP サービスのモニタリングのための **edundancy-guard-period** と **monitor-damping-period** を設定します。

これらのガードタイマーを使用して、カードの再起動およびタスクの再起動などのローカルな障害により、中断する可能性のある ICSR イベントを発生させることがないようにします。

```
configure
  context context_name
    service-redundancy-protocol variable
      guard-timer { aaa-switchover-timers { damping-period seconds |
guard-period seconds } | diameter-switchover-timers { damping-period
seconds | guard-period seconds } | srp-redundancy-timers { aaa {
damping-period seconds | guard-period seconds } | bgp { damping-period
seconds | guard-period seconds } | diam { damping-period seconds |
guard-period seconds } }
      end
```

注：

- **aaa-switchover-timers**：ネットワークのコンバージ中に AAA の障害（post ICSR switchover）が発生した場合に、バックツーバックの ICSR スイッチオーバーを防止するタイマーを設定します。
  - **damping-period**：guard-period 内のモニタリング障害による ICSR スイッチオーバーをトリガーする遅延時間を設定します。
  - **guard-period**：local-failure-recovery と network-convergence のタイマーを設定します。
- **diameter-switchover-timers**：ネットワークのコンバージ中に Diameter の障害（post ICSR switchover）が発生した場合に、バックツーバックの ICSR スイッチオーバーを防止するタイマーを設定します。
  - **damping-period**：guard-period 内のモニタリング障害による ICSR スイッチオーバーをトリガーする遅延時間を設定します。
  - **guard-period**：local-failure-recovery と network-convergence のタイマーを設定します。
- **srp-redundancy-timer**：システムがローカルの card-reboot/critical-task-restart の障害からリカバリされている間に、ICSR スイッチオーバーを防止するタイマーを設定します。
  - **aaa**：ローカル障害に続いて AAA のモニタリングの障害が発生する。
  - **bgp**：ローカル障害に続いて BGP のモニタリングの障害が発生する。
  - **diam**：ローカル障害に続いて Diameter のモニタリングの障害が発生する。

## BFD マルチホップ フォールオーバーの有効化

コンテキスト コンフィギュレーションモードの **ip route** コマンドと **ipv6 route** コマンドでの **fall-over bfd multihop *mhsess\_name*** キーワードは、指定されたマルチホップセッションのフォールオーバー BFD 機能を有効にします。**fall-over bfd** オプションは、BFD を使用してネイバーの到達可能性と活性度をモニターします。有効にすると、BFD が障害を通知した場合にセッションを切断します。

```

configure
  context context_name
    ip route { ip_address/ip_mask | ip_address ip_mask } { gateway_ip_address |
next-hop next_hop_ip_address | point-to-point | tunnel } egress_intrfc_name [
cost cost ] [ fall-over bfd multihop mhsess_name ] [ precedence precedence
] [ vrf vrf_name [ cost value ] [ fall-over bfd multihop mhsess_name ] [
precedence precedence ] +
    end

```

**Ip route** コマンドでは、静的マルチホップの BFD ルートを追加することもできます。

```

ip route static multihop bfd mhbfd_sess_name local_endpt_ipaddr
remote_endpt_ipaddr

```



**重要** SNMP トラップは、BFDセッションがアップおよびダウンしたときに生成されます (BFDSessUp と BFDSessDown)。

## ip route コマンド

```

configure
  context context_name
    ip route { ip_address/ip_mask | ip_address ip_mask } { gateway_ip_address |
next-hop next_hop_ip_address | point-to-point | tunnel } egress_intrfc_name [
cost cost ] [ fall-over bfd multihop mhsess_name ] [ precedence precedence
] [ vrf vrf_name [ cost value ] [ fall-over bfd multihop mhsess_name ] [
precedence precedence ] +
    end

```

**Ip route** コマンドでは、静的マルチホップの BFD ルートを追加することもできます。

```

ip route static multihop bfd mhbfd_sess_name local_endpt_ipaddr
remote_endpt_ipaddr

```

## ip routev6 コマンド

```

configure
  context context_name
    ipv6 route ipv6_address/prefix_length { interface name | next-hop
ipv6_address interface name } [ cost cost ] [ fall-over bfd multihop mhsess_name
] [ precedence precedence ] [ vrf vrf_name [ cost value ] [ fall-over bfd
multihop mhsess_name ] [ precedence precedence ]
    end

```

**ipv6 route** コマンドでは、スタティックマルチホップ BFD ルートも追加できるようになりました。

```

ipv6 route static multihop bfd mhbfd_sess_name local_endpt_ipv6addr
remote_endpt_ipv6addr

```

## BFD 間隔の調整

ネットワーク展開のコンバージェンス要件を満たすため、BFD パケット間の送信間隔を設定します（ミリ秒単位）。

```
configure
  context context_name
    interface interface_name broadcast
      bfd interval interval_num min_rx milliseconds multiplier value
    end
```

注：

- *milliseconds* は、50 ～ 10,000 の整数です（デフォルトは 50）

## スタンバイ ICSR シャーシからの BGP ルートのアドバタイジングの有効化

この機能の設定については、[スタンバイ ICSR シャーシからの BGP ルートのアドバタイジング \(319 ページ\)](#) をご覧ください。

## 設定の保存

「設定の確認と保存」の章の説明に従って、設定を保存します。

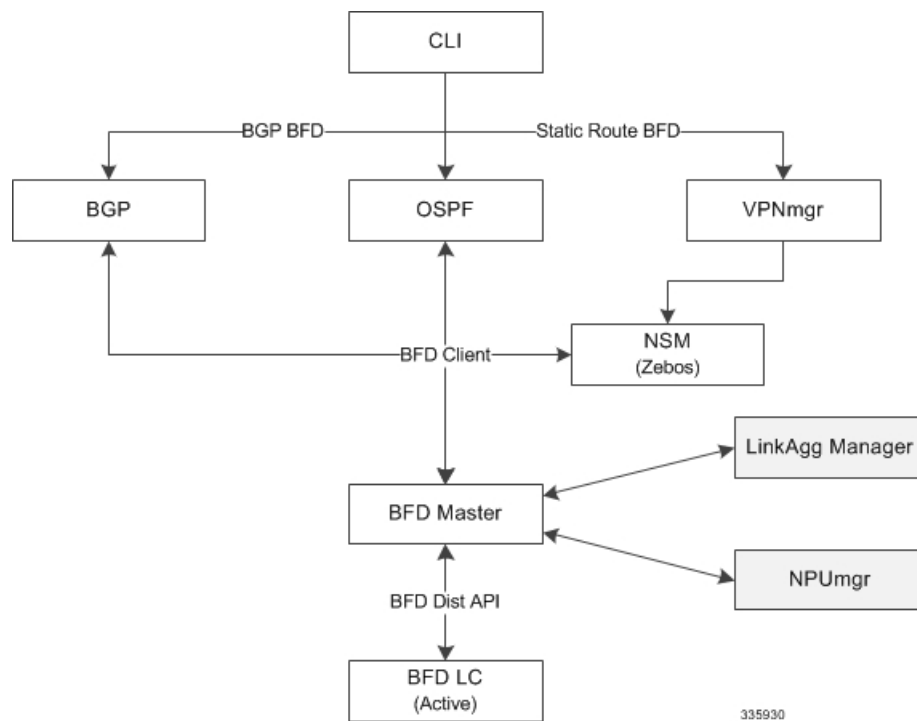
## リンク集約メンバーリンクの BFD サポート

メンバーリンクベースの BFD は、LACP よりも高速で個々のリンク障害を検出し、単一のメンバーリンク障害の結果として、セッション全体またはトラフィックのダウン期間を短縮します。

## 概要

BFD コンフィギュレーションモードの CLI コマンドは、linkagg タスクとの BFD の連携動作を設定します。セッションが設定されると、BFD はメンバーリンクごとに BFD セッションを作成し、各 linkagg メンバーリンクでパケットの送信を開始します。メンバーリンクの BFD セッションに障害が発生すると、StarOS は linkagg タスクに障害を通知します。

図 3: BFD の連携動作



スロット番号を使用して `linkagg-peer` を定義する場合は、冗長 LC (ラインカード) スロット用の `linkagg peer` を設定することもできます。その際、メンバリンク設定でスロットも指定する必要があります。また、スロットを使用せずに `linkagg-peer` を設定する場合は、スロットを指定してピアを設定する前に、そのピアを削除する必要があります。



**重要** RFC 7130 に準拠するため、`linkagg` インターフェイスごとに許可される IPv4 または IPv6 BFD セッションベースの設定は 1 つだけです。

## BFD Linkagg メンバーリンクのサポートの設定

`bfd linkagg-peer` コマンドは、メンバーリンク BFD を有効にして、BFD リンク集約 (`linkagg`) セッション値を設定します (RFC 7130)。

### configure

```

context context_name
  bfd-protocol
    bfd linkagg-peer linkagg_group_id local-endpt-addr local-endpt_ipaddress
    remote-endpt-addr remote_endpt_ipaddress interval tx_interval min_rx rx_interval
    multiplier multiplier_value [ slot slot_number ]
  no bfd linkagg-peer linkagg_group_id [ slot slot_number ]
end
  
```

注：

- **linkagg\_group\_id** は、LAG 番号を 1 ~ 255 の整数として指定します。
- **local-endpt-addr local-endpt\_ipaddress** は、IPv4 または IPv6 表記内のマルチホップ BFD セッションの送信元アドレスを指定します。
- **remote-endpt-addr remote-endpt\_ipaddress** は、IPv4 または IPv6 表記内のマルチホップ BFD セッションのリモートアドレスを指定します。
- **interval tx\_interval** は、制御パケットの送信間隔を 50 ~ 10000 の整数で（ミリ秒単位）指定します。
- **min\_rx rx\_interval** は、制御パケットの受信間隔を 50 ~ 10000 の整数で（ミリ秒単位）指定します。
- **multiplier multiplier\_value** は、ホールド時間を計算するために使用する値を 3 ~ 50 の整数で指定します。
- 冗長アクティブ/スタンバイリンク集約の **slot slot\_number** オプションは、この設定を適用するカードを指定します。

## 設定の保存

「設定の確認と保存」の章の説明に従って、設定を保存します。

## ルーティング情報の表示

現在のコンテキストのルーティング情報を表示するには、次の Exec モードのコマンドのいずれかを実行します。

- **show ip route** : 現在のコンテキストにおける IPv4 ルートの情報を表示します。
- **show ipv6 route** : 現在のコンテキストにおける ipv6 ルートの情報を表示します。
- **show ip static-route** : 現在の contextospf 内の IPv4 スタティックルートの情報のみを表示します。
- **show ip ospf** : 現在のコンテキストにおける IPv4 OSPF プロセスの概要情報を表示します。
- **show ipv6 ospf** : 現在のコンテキストにおける IPv6 OSPFv3 プロセスの概要情報を表示します。
- **show ip bgp** : IPv4 BGP 情報を表示します。

次に、**show ip route** コマンドの出力例を示します。

```
[local]host_name# show ip route
"*" indicates the Best or Used route.

Destination          Nexthop          Protocol  Prec Cost Interface
*44.44.44.0/24       208.230.231.50  static   1    0    local1
*192.168.82.0/24     0.0.0.0          connected 0    0
*192.168.83.0/24     0.0.0.0          connected 0    0
 208.230.231.0/24    0.0.0.0          ospf     110  10    local1
*208.230.231.0/24    0.0.0.0          connected 0    0    local1
Total route count: 5
```







## 第 23 章

# VLANs

この章では、強化サービスまたは拡張サービスをサポートする仮想ローカルエリアネットワーク（VLAN）の設定について説明します。製品固有や機能固有の「アドミニストレーションガイド」には、VLANを使用する可能性のあるシステムでのサービスの設定例と手順が記載されています。次に説明する手順を使用する前に、サービスモデルに最適な設定例を選択する必要があります。

- [概要](#) (337 ページ)
- [VLAN および StarOS](#) (339 ページ)
- [VLAN およびハイパーバイザ](#) (339 ページ)
- [VLAN および KVM ハイパーバイザ](#) (340 ページ)
- [VLAN 対応 VM](#) (341 ページ)
- [VLAN および VMware](#) (341 ページ)
- [VLAN タグの設定](#) (342 ページ)
- [ポート設定の確認](#) (343 ページ)
- [サブスクリバ VLAN の関連付けの設定](#) (344 ページ)
- [VLAN 関連の CLI コマンド](#) (345 ページ)

## 概要

仮想 LAN（VLAN）は、コンテキストおよびサービスの設定と使用の柔軟性を高めます。

これらはポートごとに「タグ」として設定され、より複雑な設定を実装することができます。VLAN タグを使用すると、1つの物理ポートを複数の論理インターフェイスにバインドして、異なるコンテキスト内に設定できます。したがって、各イーサネットポートは、VLAN タグが使用されている場合は多くの論理ポートが含まれているように表示されることがあります。

VLAN は、ハイパーバイザレベルと StarOS レベルで作成できます。VLAN を作成する場所は、特定のネットワーク要件によって異なります。

## オーバーラッピング IP アドレスプールのサポート : GGSN

IP アドレスプールの重複により、オペレータは、物理的に分離されたルータや仮想ルータに高価な投資をすることなく、同じプライベート IP アドレス空間を持つ複数の企業の VPN 顧客を柔軟にサポートできます。

システムは、次の 2 種類の重複プールをサポートしています。

- リソースプールはダイナミック割り当て専用に設計されており、VPN トンネル（GRE トンネルなど）を使用して、VPN との間でプライベート IP アドレスを転送および受信します。
- 重複プールは、ダイナミックアドレッシングとスタティックアドレッシングの両方に使用でき、VLAN とネクストホップ転送アドレスを使用して VPN カスタマに接続できます。

ダウンストリームトラフィックを正しい PDP コンテキストに転送するために、GGSN は GRE トンネル ID または VLAN ID のいずれかを使用してパケットを照合します。トラフィックをアップストリームに転送する場合、GGSN は IP プール設定でトンネルおよび転送情報を使用します。重複するプールは、このようなインスタンスの APN で設定する必要があります。

PDP コンテキストが作成されると、IP アドレスが IP プールから割り当てられます。この場合、転送ルールも GGSN に設定されます。アドレスが静的に割り当てられている場合、GGSN が APN で設定されたプールから IP アドレスを確認すると、転送ルールも適用されます。

GGSN は、コンテキストごとの VLAN インターフェイスと同じ数の実際の重複プールに合わせ拡張でき、GGSN ごとに複数のコンテキストが存在することがあります。制限は IP プールの数です。この拡張性により、顧客のプライベート IP アドレス空間を使用して VPN サービスを顧客に提供するオペレータが、ハードウェアコストの増加や複雑な設定について心配する必要がなくなります。

## RADIUS VLAN サポート : Enhanced Charging サービス

VPN のお客様は、多くの場合、他の顧客と簡単にオーバーラップできるプライベートアドレス空間を使用しています。サブスクリバアドレスは、同じ仮想ルーティングコンテキスト内で設定できるオーバーラッピングプールでサポートされています。

RADIUS サーバーと NAS の IP アドレスが別々のコンテキスト内である必要がないため、APN と RADIUS の設定とネットワーク設計がシンプルになります。この機能を使用すると、同じコンテキスト内に次のシナリオを定義できます。

- さまざまな APN を表すさまざまな RADIUS サーバークループの RADIUS NAS-IP アドレスのオーバーラッピング
- さまざまな RADIUS サーバークループの RADIUS サーバー IP アドレスのオーバーラッピング

オーバーラッピング NAS-IP アドレスすべてに固有のネクストホップアドレスが指定されて、固有の VLAN にバインドされるインターフェイスにバインドされます。これにより、同じコンテキストにこの設定が存在できるようになります。

システムは、RADIUS アクセス要求やアカウントメッセージを、その NAS-IP に定義されているネクストホップに転送します。接続されているルータがそのメッセージを RADIUS サーバーに転送します。ネクストホップアドレスによって、使用するインターフェイスと VLAN が決まります。サーバーからのトラフィックは、ポートや VLAN の組み合わせによって特定の NAS-IP に属しているものとして識別されます。

設定可能な RADIUS NAS-IP アドレスの数は、設定可能なループバックアドレスの数によって制限されます。

## APN サポート : PDN ゲートウェイ (P-GW)

P-GW アクセスポイント名 (APN) は、APN に対する広範なパラメータ設定の柔軟性をサポートします。APN では VLAN タグを選択できますが、APN とは別に、P-GW で設定されます。

## VLAN および StarOS

StarOS は、いくつかのゲートウェイ製品の VLAN をサポートしています。



- (注) VLAN は、サブスライバ vNIC トラフィックポート 1-10 ~ 1-21 とあわせてサポートされています。StarOS は、付録の「エンジニアリングルール」で説明されているように、VLAN の設定制限をサポートしています。

## VLAN およびハイパーバイザ

ネットワークを介して処理されているパケットのタイプに応じて、ハイパーバイザは VPC-SI 仮想マシン (VM) とパケットを交換する前に、さまざまな VLAN タスクを実行します。

- **管理パケット** MGMT パケットはタグなしで着信し、ハイパーバイザはこれらのパケットを追加の VLAN 処理なしで VM と交換します。
- **アクセスパケット** は、VLAN タグ付きで物理ネットワークから着信します。ハイパーバイザは、VLAN タグを VM に転送する前に削除します。受信したパケットは、物理ネットワークを介して送信される前に再度タグ付けされます。
- **トランキングパケット** は、VLAN タグ付きで物理ネットワークを介して発着信します。ハイパーバイザは、タグ付きのパケットを追加処理のために VM に送信する前に、タグをフィルタ処理します。

管理、アクセス、およびトランキングの各パケットは、個別のコンテキストで定義し、一意のインターフェイスにバインドする必要があります。ハイパーバイザは、適切なタイプの VLAN タグ付けを行うか、またはパケットタイプに基づいたフィルタ処理を行うように設定する必要があります。

詳細については、VLAN サポートと送信元を簡単に説明している次の項を参照してください。

- [VLAN および KVM ハイパーバイザ \(340 ページ\)](#)
- [VLAN および VMware \(341 ページ\)](#)

## VLAN および KVM ハイパーバイザ

### ネットワークの分離

Ubuntu ネットワーキングスタックの実装により、KVM ホストは、単純なレイヤ2ブリッジ（イーサネットスイッチ）、転送または NAT ルータ、ステートフルファイアウォール、またはこれらのロールの任意の組み合わせとして機能できます。

### VLAN 対ブリッジインターフェイス

KVM 仮想化のシナリオでは、VLAN 使用率をシンプルなブリッジインターフェイス共有の拡張と見なすことができます。違いは、ブリッジセットに参加するインターフェイスです。標準動作モード（イーサネットブリッジとのネットワークポート共有の例を参照）では、物理インターフェイス（eth0、eth1... など）はブリッジにバインドされ、各ゲストが使用します。これらのインターフェイスは、VLAN ID タグの有無にかかわらず、外部から着信するか、または内部で生成された未変更のパケットを伝送します。

サブインターフェイスを作成することで、特定の VLANID を伝送していないすべてのパッケージを除外できます。これらのサブインターフェイスは、特定の VLAN ID によって定義された VLAN の一部になります。

この概念をブリッジされたインターフェイスの共有方式に適用するには、バインドされた物理インターフェイスを、特定の VLAN セグメンテーションの一部であるサブインターフェイスに置き換える必要があります。このようにして、このブリッジにバインドされたインターフェイスを持つすべての仮想マシンゲストは、その特定の VLAN の一部になります。シンプルなイーサネットブリッジ環境と同様に、提供されるネットワークは透過的です。



---

(注) すべての vNIC タイプがブリッジへの VLAN トランッキングをサポートしているわけではありません。これは多くの場合、ハードウェアで VLAN が除外されたためです。

---

### その他の情報

KVM ハイパーバイザを使用した VLAN の設定の詳細については、次の URL を参照してください。

- [802.1q VLAN の設定 :](https://www.ibm.com/support/knowledgecenter/linuxonibm/liaat/liaatkvmsecconfvlans.htm)  
<https://www.ibm.com/support/knowledgecenter/linuxonibm/liaat/liaatkvmsecconfvlans.htm>

- *KVM/Networking* : <https://help.ubuntu.com/community/KVM/Networking>

## VLAN 対応 VM

VLAN 対応 VM インスタンスは、単一の vNIC を介して VLAN タグ付きトラフィックを送受信します。VLAN 対応は、VLAN タグ付きトラフィックを予期している NFV アプリケーション (VNF) に役立ち、単一の vNIC で複数のサービスを提供できます。トランクインターフェイスを備えた VLAN 対応 VM を使用すると、接続が中断されることなく、ネットワークの自動追加と削除を容易に行うことができます。

VLAN トランクは、VLAN を単一のトランクポートに結合することで、VLAN 対応インスタンスをサポートします。VLAN タグ付きトラフィックのトランクを実装するには、次の手順を実行します。

- 親ポートを作成し、新しいポートを既存の Neutron ネットワークに接続します。新しいポートを接続すると、OpenStack ネットワーキング (Neutron) は、作成した親ポートにトランク接続を追加します。
- サブポートを作成します。これらのサブポートは VLAN をインスタンスに接続してトランクへの接続を可能にします。

OpenStack ネットワーキングサービス (Neutron) がサブポートに割り当てた MAC アドレスを使用するように VM インスタンスを展開します。Elastic Services Controller (ESC) 5.8 バージョンは、この VLAN 対応 VM をサポートしています。

### 制限事項 :

SRIOV VF を介した VLAN 対応の VM に対する Red Hat での既知の RFE 障害により、VPC DI の展開時および親ポートを VM に接続する際の VLAN 対応には次の 2 つの制限があります。

- VLAN 対応 VM 設定の一部として作成された Neutron トランクポートのステータスは DOWN です。
- 複数の VLAN ID で作成されたサブポートは Detached 状態です。

## VLAN および VMware

VMware は、ネットワーク展開要件を満たすために VLAN の設定をサポートしています。

### VLAN の設定

VLAN を使用すると、単一の物理 LAN セグメントをさらに分割して、ポートのグループが物理的に異なるセグメント上にあるかのように分離することができます。

次の理由により、VLAN ありの ESXi を設定することを推奨します。

- これにより、ホストが既存の環境に統合されます。

- これにより、ホストが既存の環境に統合されます。
- これにより、ネットワークトラフィックの輻輳が軽減されます。
- iSCSI トラフィックには、隔離されたネットワークが必要です。

外部スイッチタグ (EST)、仮想スイッチタグ (VST)、および仮想ゲストタグ (VGT) の 3 つの方法で、ESXi で VLAN を設定できます。

- ESTを使用すると、パケットのすべての VLAN タグ付けが物理スイッチで実行されます。ホストネットワークアダプタは、物理スイッチのアクセスポートに接続されています。仮想スイッチに接続されているポートグループは、VLAN ID を 0 に設定する必要があります。

VSTを使用すると、ホストを離れる前にパケットのすべての VLAN タグ付けが仮想スイッチによって実行されます。ホストネットワークアダプタは、物理スイッチのトランクポートに接続されている必要があります。また、仮想スイッチに接続されているポートグループには、適切な VLAN ID が指定されている必要があります。

VGTを使用すると、すべての VLAN タグ付けが仮想マシンによって実行されます。VGT の場合、VLAN ID = 4095 です。仮想スイッチとの間でフレームが渡されると、仮想マシンのネットワークスタックと外部スイッチの間で VLAN タグが保持されます。物理スイッチポートは、トランクポートに設定されます。

## その他の情報

VMware ハイパーバイザを使用した VLAN の設定の詳細については、次のドキュメントを参照してください。

- [UCS および VMware での VLAN の設定](#)
- VLAN の設定については、[VMware のドキュメント](#)を参照してください。
- ESXi ホストへの VLAN ID の割り当てについては、[VMware のドキュメント](#)を参照してください。
- 仮想スイッチ、物理スイッチ、および仮想マシンでの VLAN の設定については、[VMware のドキュメント](#)を参照してください。

## VLAN タグの設定

ポートで VLAN を作成し、既存のインターフェイスにバインドするには、次の例を使用します。インターフェイスの作成については、「システムインターフェイスとポート」を参照してください。

```
config
  port ethernet slot/port
    no shutdown
    vlan vlan_tag_ID
    no shutdown
```

```
bind interface interface_name context_name
end
```

注：

- オプション：VLANサブスクリバの関連付けを設定します。詳細については、[サブスクリバ VLAN の関連付けの設定（344 ページ）](#) を参照してください。
- ポートの追加の VLAN を設定するには、必要に応じてこの手順を繰り返します。
- 詳細については、[VLAN 関連の CLI コマンド（345 ページ）](#) および『*Command Line Interface Reference*』を参照してください。
- 「設定の確認と保存」の章の説明に従って、設定を保存します。

## ポート設定の確認

次のコマンドを実行してポート設定を確認します。

```
[local]host_name# show port info slot/port
```

ポートに対して少なくとも 1 つの VLAN が設定されている場合のこのコマンドの出力例を次に示します。

```
Port: 5/11
Port Type           : 10G Ethernet
Role                : Service Port
Description         : (None Set)
Redundancy Mode     : Port Mode
Redundant With      : 6/11
Preferred Port      : Non-Revertive
Physical ifIndex    : 85262336
Administrative State : Enabled
Configured Duplex   : Auto
Configured Speed    : Auto
Fault Unidirection Mode : 802_3ae clause 46
Configured Flow Control : Enabled
Interface MAC Address : 64-9E-F3-69-5B-EA
SRP Virtual MAC Address : None
Fixed MAC Address   : 64-9E-F3-69-5B-CA
Link State          : Up
Link Duplex         : Full
Link Speed          : 10 Gb
Flow Control        : Enabled
Link Aggregation Group : None
Untagged:
  Logical ifIndex   : 85262337
  Operational State : Up, Active
Tagged VLAN: VID 10
  Logical ifIndex   : 285278210
  VLAN Type         : Standard
  VLAN Priority     : 0
  Administrative State : Enabled
  Operational State : Up, Active
Number of VLANs    : 1
SFP Module         : Present (10G Base-SR)
```

注：

- 必要に応じてこの手順を繰り返して、追加ポートを確認します。
- オプション：VLANサブスクリバの関連付けを設定します。詳細については、[サブスクリバ VLAN の関連付けの設定 \(344 ページ\)](#) を参照してください。
- 追加情報については、[VLAN 関連の CLI コマンド \(345 ページ\)](#) を参照してください。
- 「設定の確認と保存」の章の説明に従って、設定を保存します。

## サブスクリバ VLAN の関連付けの設定

サブスクリバトラフィックは、ユーザープロファイルの設定に基づいて特定の VLAN にルーティングできます。この機能によって、指定した VLAN を介してサブスクリバからすべてのトラフィックをルーティングするメカニズムが実現します。また、サブスクリバ宛のすべてのパケットは VLAN 上で有効な IP アドレスのみを使用して送信する必要があります。そうでない場合はドロップされます。

## 使用される RADIUS 属性

次の RADIUS 属性を RADIUS サーバー上のサブスクリバプロファイル内で設定して、特定の VLAN をサブスクリバに関連付けることができます。

- **SN-Assigned-VLAN-ID** : Starent VSA ディクショナリ内
- **SN1-Assigned-VLAN-ID** : Starent VSA1 ディクショナリ内



**重要** サブスクリバプロファイルの設定手順は RADIUS サーバーアプリケーション間で異なるため、この項では、サブスクリバプロファイルに追加できる個々の属性についてのみ説明します。サブスクリバの設定手順については、RADIUS サーバーに付属のマニュアルを参照してください。

## ローカルサブスクリバのプロファイルの設定

システム上のローカルサブスクリバのプロファイル内で VLAN の関連付けを設定するには、次の設定例を使用します。



**重要** これらの手順は、[VLAN タグの設定 \(342 ページ\)](#) に記載されている手順に従って、サブスクリバタイプの VLAN タグがすでに設定されていることを前提としています。

```
config
context context_name
subscriber name user_name
```



```
ip vlan vlan_id
end
```

## サブスクリバのプロファイル設定の確認

サブスクリバプロファイルの設定を表示するには、次のコマンドを使用します。

```
[local]host_name# show subscriber configuration username user_name
```

注：

- 各サブスクリバに対してこのコマンドを繰り返します。
- 「設定の確認と保存」の章の説明に従って、設定を保存します。

## VLAN 関連の CLI コマンド

VLAN 関連の機能は、いくつかの CLI コマンドモードでサポートされています。次の表に、VLAN 関連の機能の設定とモニタリングに関連するコマンドを示します。

次に示すコマンドの使用に関する詳細については、『*Command Line Interface Reference*』を参照してください。

表 23: VLAN 関連の設定コマンド

CLI モード	コマンド	説明
AAA サーバー グループ コンフィギュレーション モード	<b>radius attribute nas-ip-address</b> <b>address</b> <i>ip_address</i> <b>nexthop-forwarding-address</b> <i>ip_address</i> <b>vlan</b> <i>vlan_id</i>	単一ネクストホップゲートウェイモードで実行しているシステムへのネクストホップ転送アドレスを持つ VLAN ID を提供するように RADIUS クライアントを設定します。  注： <b>vlan</b> キーワードにアクセスするには、グローバルコンフィギュレーションモードで <b>aaa-large configuration</b> を有効にする必要があります。
ACS 課金アクションコンフィギュレーション モード	<b>ip vlan</b> <i>vlan_id</i>	接続先コンテキストのサブスクリバトラフィックに関連付ける VLAN 識別子を設定します。

CLI モード	コマンド	説明
コンテキスト コンフィギュレーション モード	<b>ip pool</b> <i>pool_name</i> <b>nexthop forwarding address</b> <i>ip_address</i> <b>overlap vlanid</b> <i>vlan_id</i>	テキストホップ転送アドレスが設定されている場合、 <b>overlap vlanid</b> を使用すると IP アドレスプールのオーバーラップが可能になり、指定した VLAN ID にプールを関連付けます。
コンテキスト コンフィギュレーション モード	<b>ip routing overlap-pool</b>	VLAN ID を使用してプールの重複が設定されている場合、ダイナミックルーティングプロトコルでオーバーラッププールアドレスがアドバタイズされます。有効にすると、オーバーラップアドレスがインターフェイスアドレスとして追加され、アドバタイズされます。
コンテキスト コンフィギュレーション モード	<b>radius attribute nas-ip-address address</b> <i>ip_address</i> <b>nexthop-forwarding-address</b> <i>ip_address</i> <b>vlan</b> <i>vlan_id</i>	ネクストホップ IP アドレスに関連付ける VLAN ID を指定します。
イーサネット インターフェイス コンフィギュレーション モード	<b>[no] logical-port-statistics</b>	最初の32設定されたイーサネットまたはPVCインターフェイスタイプの論理ポート（VLANとNPU）のバルク統計情報の収集を有効または無効にします。
イーサネット インターフェイス コンフィギュレーション モード	<b>vlan-map next-hop</b> <i>ipv4_address</i>	単一ネクストホップ IP アドレスを設定して、複数の VLAN が単一のネクストホップゲートウェイを使用できるようにします。VLAN マップが特定のインターフェイスに関連付けられます。
イーサネットポート コンフィギュレーション モード	<b>vlan</b> <i>vlan_id</i>	VLAN コンフィギュレーション モードを開始します。
PVC コンフィギュレーション モード	<b>[no] shutdown</b>	指定した VLAN 上のトラフィックを有効または無効にします。次を参照してください。

CLI モード	コマンド	説明
サブスクライバコンフィギュレーションモード	<b>ip vlan</b> <i>vlan_id</i>	サブスクライバセッションがパケットを受信するために割り当てられたアドレスとともに使用するサブスクライバ VLAN ID を設定します。アドレスが割り当てられている IP プールが VLAN ID を使用して設定されている場合、このサブスクライバで設定された VLAN ID がそのアドレスを上書きします。
VLAN コンフィギュレーションモード	<b>bind interface</b> <i>interface_name</i> <i>context_name</i>	VLAN サービスをサポートするために、仮想インターフェイスとコンテキストをバインドします。
VLAN コンフィギュレーションモード	<b>[no] ingress-mode</b>	ポート入力着信モードを有効または無効にします。
VLAN コンフィギュレーションモード	<b>priority</b> <i>value</i>	ASN-GW サービスの 802.1p VLAN 優先順位ビットのみを設定します。
VLAN コンフィギュレーションモード	<b>[no] shutdown</b>	現在の VLAN 上のトラフィックを有効または無効にします。
VLAN コンフィギュレーションモード	<b>vlan-map interface</b> <i>if_name</i> <i>context_name</i>	VLAN ID を持つ IP インターフェイスをコンテキストに関連付けます。

表 24: VLAN 関連のモニタリングコマンド

CLI モード	コマンド	説明
Exec モードの show コマンド	<b>clear port</b> <i>slot/port</i> <b>vlan</b> <i>vlan_id</i>	以前に設定された VLAN ID を持つポートの NPU 統計をクリアします。
Exec モードの show コマンド	<b>show logical-port utilization table</b> <b>vlan { 5-minute   hourly }</b>	指定された収集間隔の VLAN 使用率を表示します。
Exec モードの show コマンド	<b>show port info</b> <i>slot/port</i> <b>vlan</b> <i>vlan_id</i>	以前に設定された VLAN ID の NPU カウンタを表示します。





## 第 24 章

# BGP MPLS VPN

この章では、ボーダーゲートウェイプロトコル (BGP)、マルチプロトコルラベルスイッチング (MPLS)、バーチャルプライベートネットワーク (VPN) でサポートされるサービスについて説明します。



**重要** MPLSは、別途ライセンスを必要とするライセンス供与されたシスコの機能です。特定のライセンス要件の詳細については、シスコのアカウント担当者にお問い合わせください。ライセンスのインストールと確認の詳細については、「ソフトウェア管理操作」の「ライセンスキーの管理」の項を参照してください。

説明する項目は次のとおりです。

- [はじめに \(349 ページ\)](#)
- [PE に接続された MPLS-CE \(350 ページ\)](#)
- [VPC-SI PE として \(350 ページ\)](#)
- [BGP MPLS VPN の IPv6 サポート \(352 ページ\)](#)
- [VPN 関連の CLI コマンド \(356 ページ\)](#)

## はじめに

サービスプロバイダーは、多数の異なるアドレッシングモデルと要件を持つ多数の企業のアクセスポイント名 (APN) をサポートする能力を必要とします。VPC-SIはBGP MPLS レイヤ3 VPNを使用して、拡張性の高い方法で企業顧客のAPNを分離します。このソリューションは、RFC4364 (BGP/MPLS IPのバーチャルプライベートネットワーク (VPN) に準拠しています。



(注) オプション b は、マルチ AS バックボーン間の接続でサポートされています。

BGP/MPLS ソリューションでは、次のシナリオがサポートされています。

- [PE に接続された MPLS-CE \(350 ページ\)](#)

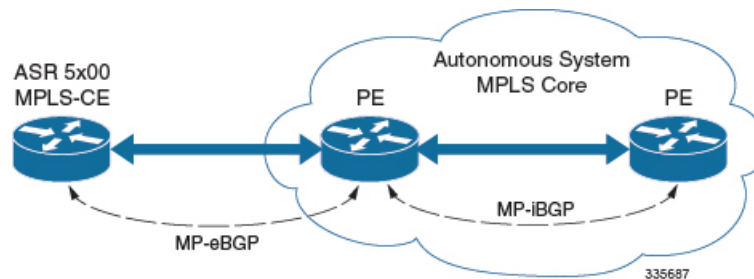
- VPC-SI PE として (350 ページ)

また、VPC-SI は RFC 4659 (*BGP-MPLS IP Virtual Private Network (VPN) Extension for IPv6 VPN*) で説明されているとおり、VPNv6 をサポートします。詳細については、[BGP MPLS VPN の IPv6 サポート \(352 ページ\)](#) を参照してください。

## PE に接続された MPLS-CE

このシナリオでは、VPC-SI はプロバイダーエッジ (PE) ラベルエッジルータ (LER) に接続されている MPLS-CE (カスタマーエッジ) ネットワーク要素として機能し、これにより MPLS コア (RFC 4364) に接続されます。以下の図を参照してください。

図 4: VPC-SIMPLS-CE から PE



MPLS-CE は、独自の自律システム (AS) 内の PE ルータのように機能します。これにより、Virtual Route Forwarding (VRF) ルートが維持され、MP eBGP (マルチプロトコル外部 BGP) セッションを介して、VPN ルート情報が PE と交換されます。

PE も VRF を使用して設定され、MP-iBGP (マルチプロトコル内部 BGP) 接続を介して AS 内の他の PE と VPN ルートを交換し、MP-eBGP 接続を介して MPLS-CE を交換します。

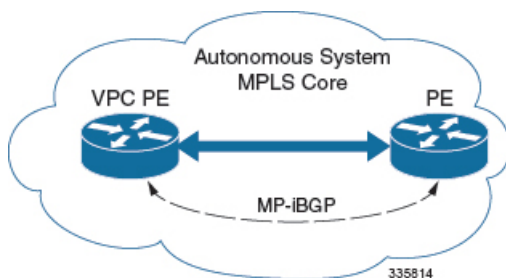
EBGP 接続を使用すると、PE は、IBGP ピアから学習したルート内のネクストホップ IP アドレスとラベルを変更してから、MPLS-CE にアドバタイズすることができます。この場合、MPLS-CE は、ルートをアドバタイズして学習するために、MP-eBGP だけを使用します。直接接続 EBGP ピアリングのため、Label Distribution Protocol (LDP) および Resource Reservation Protocol (RSVP) は必要ありません。このシナリオの MPLS-CE は、PE との間で (MP-eBGP 接続を介して学習された) 単一のラベルをプッシュ/ポップします。

## VPC-SI PE として

### 概要

このシナリオでは、VPC-SI は MPLS コアのエッジにある PE ルータとして機能します。以下の図を参照してください。

図 5: VPC-SI PE として



VPC-SI では、最初の 2 つのシナリオに示すように、ASBR または PE の必要性を排除します。このシナリオでは、IBGP 機能と MPLS ラベル配布プロトコルという 2 つの主要な要件が導入されています。

VPC-SI は、次の 2 つのラベルを追加するように設定できます。

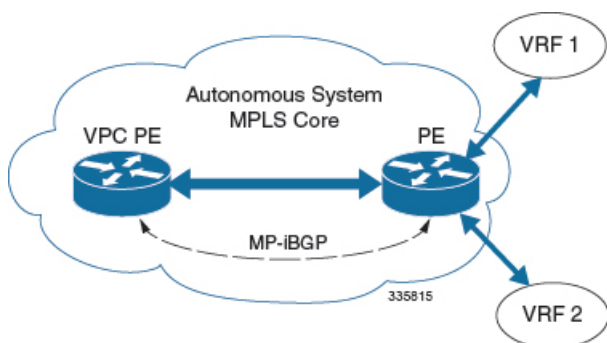
- LDP または RSVP TE から学習した外部ラベル（RSVP トラフィックエンジニアリング）
- MP-iBGP から学習した内部ラベル

このソリューションは、VPC-SI を介して開始されたトラフィック エンジニアリングと QoS をサポートします。

## 設定例

この例では、VRF は ASR 5500 PE で設定され、プールは VRF に関連付けられています。VPC-SI は、VPN ルートを IBGP ピア（PE ルータ）と交換し、LDP 経由で PE に到達するための MPLS パスを学習します。VPC-SI は、2 つのラベル（PE から学習した内部ラベルとネクストホップ IBGP ネイバーから学習した外部ラベル）を持つネクストホップにパケットを転送します。

図 6: 設定例



```
mpls ip
  protocol ldp
    enable
  exit
exit

ip vrf vrf1
  mpls traffic-class copy
```

```
exit
ip vrf vrf2
  mpls traffic-class value 5
exit

router bgp 300
  ip vrf vrf1
    route-target export 300 1
    route-target import 300 1
    route-distinguisher 300 1
  exit
  ip vrf vrf2
    route-target export 300 2
    route-target import 300 2
    route-distinguisher 300 2
  exit

router-id 2.2.2.2
neighbor 192.168.107.20 remote-as 300
neighbor 192.168.107.20 update-source nodel_loopback

address-family vpnv4
  neighbor 192.168.107.20 activate
  neighbor 192.168.107.20 send-community both
  neighbor 192.168.107.20 next-hop-self
exit

address-family ipv4 vrf vrf1
  redistribute connected
exit

address-family ipv4 vrf vrf2
  redistribute connected
exit

interface interface_to_internet
  ip address 192.168.109.65/24
  mpls ip
exit
router ospf
  network 192.168.109.0/24 area 0.0.0.0
exit
```

## BGP MPLS VPN の IPv6 サポート

### 概要

VPC-SI は RFC 4659 (*BGP-MPLS IP Virtual Private Network (VPN) Extension for IPv6 VPN*) で説明されているとおり、VPNv6 をサポートします。

IPv6 VPN は、PE ルータを通じ、IPv6 インターフェイスまたはサブインターフェイスを介してサービスプロバイダー (SP) のバックボーンに接続されます。このサイトは IPv4 と IPv6 の両方に対応できます。各 VPNv6 には独自のアドレス空間があります。つまり、特定のアドレスが異なる VPN 内の異なるシステムを示すこととなります。これは、ルート識別子 (RD) を IP アドレスに付加する VPNv6 アドレスファミリによって実現します。



VPNv6 アドレスのバイト数は 24 で、8 バイトの RD から始まり、16 バイトの IPv6 アドレスで終わります。サイトが IPv4 と IPv6 対応の場合、同じ RD を IPv4 と IPv6 の両方のアドレスのアドバタイズメントに使用できます。

システムは、IPv6 ルートに RD を追加し、VPNv6 アドレスファミリーを使用してラベル付けされた IPv6 を交換します。VPNv6 ルートのアドレスファミリー識別子 (AFI) と後続のアドレスファミリー識別子 (SAFI) のフィールドは、2 と 128 にそれぞれ設定されます。

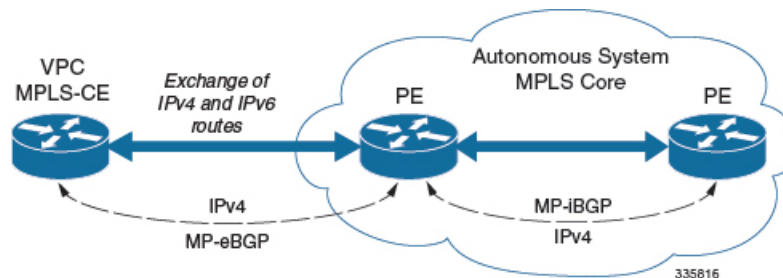
IPv6 VPN トラフィックは、IPv4 トンネリングを介して BGP スピーカーに転送されます。BGP スピーカーは、8 オクテットの RD がゼロに設定され、16 オクテットの IPv6 アドレスがアドバタイズルータの IPv4 アドレスを含む IPv4 マッピング IPv6 アドレス (RFC 4291) としてエンコードされている VPN-IPv6 アドレスを含むネクスト ホップ ネットワーク アドレス フィールドをピアにアドバタイズします。これは、VPNv6 ルートを交換するために EBGP ピアリングのみが使用されることを前提としています。

VPN IPv6 のサポートは、次を前提としています。

- デュアルスタック (IPv4/IPv6) ルーティング
- VRF の IPv6 プール
- 直接接続された IPv4 インターフェイスを介した BGP ピアリング

以下の図を参照してください。

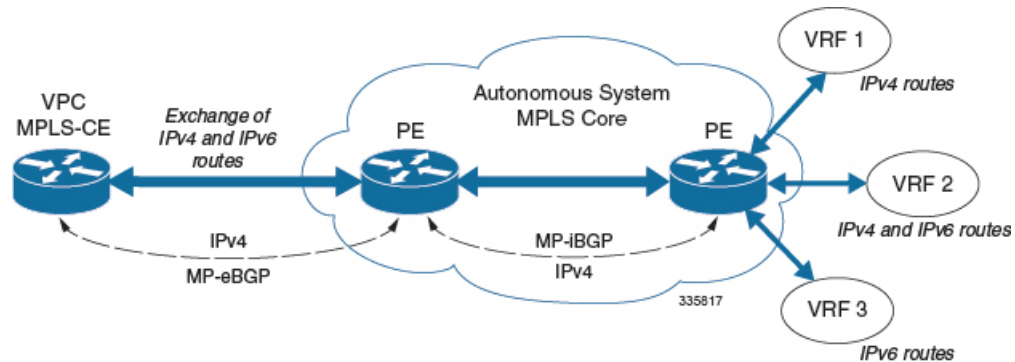
図 7: VPNv6 に対する IPv6-RD サポート



## 設定例

この例では、3 つの VRF を想定しています。VRF 1 には IPv4 ルートのみがあり、VRF f2 には IPv4 と IPv6 の両方のルートがあり、VRF 3 には IPv6 ルートのみがあります。

図 8: VPNv6 の設定例



VRF を設定します。

```
ip vrf vrf1
exit
ip vrf vrf2
exit
ip vrf vrf3
exit
```

MPLS bgp forwarding を有効にします。

```
mpls bgp forwarding
```

プールを設定します。

```
ip pool vrf1-pool 51.52.53.0 255.255.255.0 private 0 vrf vrf1
exit
ip pool vrf2-pool 51.52.53.0 255.255.255.0 private 0 vrf vrf2
exit
ipv6 pool vrf2-v6pool prefix 2005:0101::/32 private 0 vrf vrf2
exit
ipv6 pool vrf3-v6pool prefix 2005:0101::/32 private 0 vrf vrf3
exit
```

インターフェイスを設定します。

```
interface ce_interface_to_rtr
ip address 192.168.110.90 255.255.255.0
exit
interface ce_v6_interface
ip address 2009:0101:0101:0101::1/96
exit
interface ce_loopback loopback
ip address 52.1.2.3 255.255.255.255
exit
interface vrf1-loop loopback
ip vrf forwarding vrf1
ip address 1.52.53.54 255.255.255.255
exit
interface vrf2-loop loopback
ip vrf forwarding vrf2
ip address 2.52.53.54 255.255.255.255
exit
interface vrf2-v6loop loopback
ip vrf forwarding vrf2
ip address 2005:0202:0101::1/128
exit
```

```
interface vrf3-v6loop loopback
 ip vrf forwarding vrf3
 ip address 2005:0303:0101::1/128
 exit
```

アドレスファミリーおよび再配布ルールとともに BGP を設定します。

```
router bgp 800
 router-id 1.1.1.1
 neighbor 192.168.110.20 remote-as 1003
 neighbor 192.168.110.20 activate
 address-family vpnv4
 neighbor 192.168.110.20 activate
 neighbor 192.168.110.20 send-community both
 exit
 address-family vpnv6
 neighbor 192.168.110.20 activate
 neighbor 192.168.110.20 send-community both
 exit
 ip vrf vrf1
 route-distinguisher 800 1
 route-target export 800 1
 route-target import 800 1
 exit
 address-family ipv4 vrf vrf1
 redistribute connected
 redistribute static
 exit
 ip vrf vrf2
 route-distinguisher 800 2
 route-target export 800 2
 route-target import 800 2
 exit
 address-family ipv4 vrf vrf2
 redistribute connected
 redistribute static
 exit
 address-family ipv6 vrf vrf2
 redistribute connected
 redistribute static
 exit
 ip vrf vrf3
 route-distinguisher 800 3
 route-target export 800 3
 route-target import 800 3
 exit
 address-family ipv6 vrf vrf3
 redistribute connected
 redistribute static
 exit
```

APN を設定します。

```
apn walmart51.com
 selection-mode sent-by-ms
 accounting-mode none
 aaa group walmart-group
 authentication pap 1 chap 2 allow-noauth
 ip context-name Gi_ce
 ip address pool name vrf1-pool
 exit
 apn amazon51.com
 selection-mode sent-by-ms
 accounting-mode none
 aaa group amazon-group
```

```

authentication pap 1 chap 2 allow-noauth
ip context-name Gi_ce
ip address pool name vrf2-pool
ipv6 address prefix-pool vrf2-v6pool
exit
apn apple51.com
selection-mode sent-by-ms
accounting-mode none
aaa group apple-group
authentication pap 1 chap 2 allow-noauth ip context-name Gi_ce
ipv6 address prefix-pool vrf3-v6pool
exit
aaa-group amazon-group
radius ip vrf vrf2
aaa group default
exit
gtp group default
exit
ip igmp profile default
exit

```

物理インターフェイスをポートにバインドします。

## VPN 関連の CLI コマンド

VPN 関連の機能は、いくつかの CLI コマンドモードでサポートされています。次の表は、VPN 関連の機能の設定とモニタリングに関連するコマンドを示しています。

次に示すコマンドの使用に関する詳細については、『*Command Line Interface Reference*』を参照してください。

表 25: VPN 関連の設定コマンド

CLI モード	コマンド	説明
BGP アドレスファミリー (IPv4/IPv6) コンフィギュレーションモード	<b>neighbor ip_address activate</b>	ピアルータとのルーティング情報の交換を有効にします。
BGP アドレスファミリー (IPv4/IPv6) コンフィギュレーションモード	<b>neighbor ip_address send community { both   extended   standard }</b>	ピアルータ (ネイバー) にコミュニティ属性を送信します。
BGP アドレスファミリー (IPv4/IPv6) コンフィギュレーションモード	<b>redistribute connected</b>	別のプロトコルから BGP へのルートを BGP ネイバーとして再配布します。
BGP アドレスファミリー (VPNv4) コンフィギュレーションモード	<b>neighbor ip_address activate</b>	ピアルータとのルーティング情報の交換を有効にします。

CLI モード	コマンド	説明
BGP アドレスファミリー (VPNv4) コンフィギュレーション モード	<b>neighbor ip_address send community { both   extended   standard }</b>	ピアルータに拡張コミュニティ属性を送信します。VPNでは、ルート識別子とルートターゲットはBGP 拡張コミュニティでエンコードされます。このコマンドは、拡張コミュニティを持つBGP ルーをネイバーに送信できるようにします。
BGP アドレスファミリー (VRF) コンフィギュレーション モード	<b>neighbor ip_address activate</b>	ピアルータとのルーティング情報の交換を有効にします。
BGP アドレスファミリー (VRF) コンフィギュレーション モード	<b>neighbor ip_address send community { both   extended   standard }</b>	ピアルータに拡張コミュニティ属性を送信します。VPNでは、ルート識別子とルートターゲットはBGP 拡張コミュニティでエンコードされます。このコマンドは、拡張コミュニティを持つBGP ルーをネイバーに送信できるようにします。
BGP アドレスファミリー (VRF) コンフィギュレーション モード	<b>redistribute connected</b>	別のプロトコルから BGP へのルートを BGP ネイバーとして再配布します。
BGP コンフィギュレーション モード	<b>address-family { ipv4 vrf vrf_name   vpnv4 }</b>	IPv4 VRF のルーティング情報の交換を有効にします。アドレスファミリーごとに異なるモードがあります。
BGP コンフィギュレーション モード	<b>address-family { ipv6 vrf vrf_name   vpnv6 }</b>	BGP で VPNv6 アドレスファミリーと IPv6 VRF ルーティングを設定します。
BGP コンフィギュレーション モード	<b>ip vrf vrf_name</b>	BGP に VRF を追加し、VRF コンフィギュレーション モードにスイッチして、VRF の BGP 属性を設定できるようにします。
BGP IP VRF コンフィギュレーション モード	<b>route-distinguisher { as_value   ip_address } rd_value</b>	VRF のルート識別子 (RD) を割り当てます。RD 値は、VRF ごとにルータ上の一意の値にする必要があります。

CLI モード	コマンド	説明
BGP IP VRF コンフィギュレーション モード	<b>route-target</b> { <b>both</b>   <b>import</b>   <b>export</b> } { <i>as_value</i>   <i>ip_address</i> } <i>rt_value</i>	インポートおよびエクスポートのルートターゲット拡張コミュニティのリストを VRF に追加します。
コンテキスト コンフィギュレーション モード	<b>ip pool</b> <i>pool_name</i> <i>addr_range</i> <b>vrf</b> <i>vrf_name</i> [ <b>mpls-label</b> <b>input</b> <i>inlabel1</i> <b>output</b> <i>outlabel1</i> <i>outlabel2</i> ]	指定された VRF にプールを設定します。このパラメータは、Next-Hop パラメータとともに指定する必要があります。 <i>inlabel1</i> は、このプールを宛先とする着信トラフィックを識別する MPLS ラベルです。 <i>outlabel1</i> と <i>outlabel2</i> は、このプールからサブスクライバに送信されるパケットに追加する MPLS ラベルを指定します。
コンテキスト コンフィギュレーション モード	<b>ip vrf</b> <i>vrf_name</i>	VRF を作成し、VRF-ID を割り当てます。VRF がルータに作成されます。
コンテキスト コンフィギュレーション モード	<b>ipv6 pool</b> <i>pool_name</i> <b>vrf</b> <i>vrf_name</i>	プールを VRF に関連付けます。 注：デフォルトでは、設定された ipv6 プールはグローバルルーティング ドメインに関連付けられません。
コンテキスト コンフィギュレーション モード	<b>mpls bgp forwarding</b>	MPLS のボーダーゲートウェイプロトコル (BGP) 転送をグローバルに有効化します。

CLI モード	コマンド	説明
コンテキスト コンフィギュレーション モード	<b>mpls exp value</b>	3 ビット MPLS EXP ヘッダーのデフォルトの動作をベストエフォートとして設定します。この値は、コンテキスト内のすべての VRF に適用されます。デフォルトの動作では、DSCP から EXP への明示的な設定がない場合に、モバイルサブスクライバのトラフィックの DSCP 値が EXP ヘッダーにコピーされます ( <b>mpls map-dscp-exp dscp n exp m</b> コマンドを使用)。  <b>mpls exp</b> はデフォルトの動作を無効にし、EXP 値を設定された値に設定します。
コンテキスト コンフィギュレーション モード	<b>mpls ip</b>	通常ルーティングされるパスに沿って IPv4 パケットの MPLS 転送がグローバルに行われるようにします。
コンテキスト コンフィギュレーション モード	<b>radius change-authorize-nas-ip ip_address ip_address { encrypted   key } value port port_num mpls input inlabel output outlabel1 outlabel2</b>	指定された MPLS ラベルを使用するように COA トラフィックを設定します。 <i>inlabel</i> は、着信 COA トラフィックを識別します。 <i>outlabel1</i> と <i>outlabel2</i> は、COA 応答に追加する MPLS ラベルを指定します。 <i>outlabel1</i> は内部出力ラベル、 <i>outlabel2</i> は外部出力ラベルです。
イーサネット インターフェイス コンフィギュレーション モード	<b>mpls ip</b>	このインターフェイスで IP パケットのダイナミック MPLS 転送を無効にします。
Exec モード	<b>clear ip bgp peer</b>	BGP セッションをクリアします。
Exec モード	<b>lsp-ping ip_prefix_FEC</b>	指定された転送等価クラス (FEC) の MPLS ラベルスイッチドパス (LSP) 接続を確認します。その後、IPv4 または IPv6 の FEC プレフィックスが続く必要があります。

CLI モード	コマンド	説明
Exec モード	<b>lsp-traceroute</b> <i>ip_prefix_FEC</i>	パケットが宛先に転送されるときに実際にたどる MPLS LSP ルートを検出します。その後には IPv4 または IPv6 の FEC プレフィックスが続く必要があります。
IPVRF コンテキスト コンフィギュレーション モード	<b>mpls map-dscp-to-exp</b> <b>dscp</b> <i>dscp_bit_value</i> <b>exp</b> <i>exp_bit_value</i>	IP パケットヘッダーにおける最終の Differentiated Services Code Point (DSCP; DiffServ コードポイント) ビット値を、着信トラフィックの MPLS ヘッダーにおける最終の Experimental (EXP) ビット値にマッピングします。
IPVRF コンテキスト コンフィギュレーション モード	<b>mpls map-exp-to-dscp</b> <b>exp</b> <i>exp_bit_value</i> <b>dscp</b> <i>dscp_bit_value</i>	MPLS ヘッダーの着信 EXP ビット値を発信トラフィックの IP パケットヘッダーの内部 DSCP ビット値にマッピングします。
MPLS-IP コンフィギュレーション モード	<b>protocol</b> <b>ldp</b>	MPLS プロトコルファミリのコンフィギュレーションモードを作成するか、または既存のプロトコルを設定して、現在のコンテキストで MPLS-LDP コンフィギュレーションモードを開始します。このコマンドは、MPLS プロトコルファミリのプロトコルパラメータを設定します。
MPLS-LDP コンフィギュレーション モード	<b>advertise-labels</b> { <b>explicit-null</b>   <b>implicit-null</b> }	このコンテキストでシステムによってアドバタイズされたすべてのプレフィックスについて、暗黙的ヌルまたは明示的ヌルラベルのアドバタイズメントを設定します。
MPLS-LDP コンフィギュレーション モード	<b>discovery</b> { <b>hello</b> { <b>hello-interval</b> <i>seconds</i>   <b>hold-interval</b> <i>seconds</i> }   <b>transport-address</b> <i>ip_address</i> }	Label Distribution Protocol (LDP; ラベル配布プロトコル) のネイバー探索パラメータの設定
MPLS-LDP コンフィギュレーション モード	<b>enable</b>	ラベル配布プロトコル (LDP) を有効にします。
MPLS-LDP コンフィギュレーション モード	<b>router-id</b> <i>ip_address</i>	LDP ルータ ID を設定します。



CLI モード	コマンド	説明
MPLS-LDP コンフィギュレーションモード	<b>session timers { hold-interval <i>seconds</i>   keepalive-interval <i>seconds</i> }</b>	LDPセッションパラメータを設定します。

表 26: VPN 関連のモニタリングコマンド

CLI モード	コマンド	説明
Exec モードの show コマンド	<b>show ip bgp neighbors</b>	BGP ネイバーに関する情報を表示します。
Exec モードの show コマンド	<b>show ip bgp vpnv4 { all   route-distinguisher   vrf }</b>	すべての VPNv4 ルーティングデータ、VRF またはルート識別子のルーティングデータを表示します。
Exec モードの show コマンド	<b>show ip bgp vpnv6</b>	VPNv6 ルーティングテーブルの内容を表示します。
Exec モードの show コマンド	<b>show ip bgp vpnv6 { all   route-distinguisher   vrf }</b>	すべての VPNv6 ルーティングデータ、VRF またはルート識別子のルーティングデータを表示します。
Exec モードの show コマンド	<b>show ip pool</b>	設定された VRF を含むプールの詳細を表示します。
Exec モードの show コマンド	<b>show mpls cross-connect</b>	MPLS 相互接続情報を表示します。インターフェイスとラベルスイッチドパス (LSP) 間で相互接続する MPLS トンネルは、LSP をコンジットとして使用する MPLS トンネルを介して、同じタイプの 2 つの遠隔インターフェイス回線を接続します。
Exec モードの show コマンド	<b>show mpls ftm [ vrf <i>vrf_name</i> ]</b>	MPLS FEC-to-NHLFE (FTN) テーブルの情報を表示します。
Exec モードの show コマンド	<b>show mpls ftm [ vrf <i>vrf_name</i> ]</b>	指定された VRF の MPLS FTN テーブルの内容を表示します。
Exec モードの show コマンド	<b>show mpls ilm</b>	MPLS の着信ラベルマップ (ILM) テーブルの情報を表示します。
Exec モードの show コマンド	<b>show mpls ldp</b>	MPLS LDP 情報を表示します。

CLI モード	コマンド	説明
Exec モードの show コマンド	<b>show mpls nexthop-label-forwarding-entry</b>	MPLS のネクストホップラベル転送エントリ (NHLFE) テーブルの情報を表示します。



## 第 25 章

# コンテンツサービスのステアリング

この章では、コンテンツ サービス ステアリング (CSS) の設定について説明します。製品アドミニストレーションガイドには、システム上での基本サービスの設定例と手順が示されています。以下で説明する手順を実行する前に、それぞれの製品管理ガイドの説明に従って、サービスモデルに最適な設定例を選択し、そのモデルに必要な要素を設定する必要があります。



**重要** 内部 CSS は汎用的な機能で、ECSv2 ライセンスがシステムにインストールされている場合は、内部 CSS を有効にできます。内部 CSS を有効にするために別のライセンスを使用する必要はありません。ライセンスの入手方法の詳細については、最寄りのシスコのアカウント担当者にお問い合わせください。

この章は次のトピックで構成されています。

- [概要 \(363 ページ\)](#)
- [内部コンテンツサービスのステアリングの設定 \(364 ページ\)](#)

## 概要

Content Server Selection (CSS) は、モバイルサブスクリバによって提供されるか、モバイルサブスクリバに送信されるデータの「コンテンツ」に基づいてトラフィックを処理する方法を定義する StarOS 機能です。CSS は、ロードバランシング、NAT、HTTP リダイレクション、DNS リダイレクションなどの機能を含む広範な用語です。

コンテンツサーバー (サービス) は、プラットフォームの外部に置くことも、プラットフォームの内部に統合することもできます。

CSS は、アクセスコントロールリスト (ACL) を使用して、サブスクリバトラフィックフローをリダイレクトします。ACL は、システムに出入りするパケットのフローを制御します。ACL は、フィルタ基準に一致するパケットに対して実行されるアクションを制御する「ルール」(ACL ルール) またはフィルタで構成されます。

ACL はコンテキストごとに設定可能であり、サブスクリバプロファイル (または宛先コンテキストの APN プロファイル) を通じてサブスクリバに適用されます。詳細については、「アクセスコントロールリスト」の章を参照してください。

## 内部コンテンツサービスのステアリングの設定

すべてのサブスクリバの IP トラフィックを内部のインラインサービスにリダイレクトするための単一 CSS サービスを設定しアクティブ化するには、次のようにします。

- 
- ステップ1 内部 CSS の IP アクセスリストの定義 (364 ページ) の説明に従って、IP ACL を定義します。
  - ステップ2 オプション: 個々のサブスクリバへの ACL の適用 (オプション) (365 ページ) の説明に従って、個々のサブスクリバに ACL を適用します。
  - ステップ3 オプション: 複数のサブスクリバへの ACL の適用 (オプション) (365 ページ) の説明に従って、複数のサブスクリバに単一の ACL を適用します。
  - ステップ4 オプション: 複数のサブスクリバへの APN を介した ACL の適用 (296 ページ) の説明に従って、APN を介して複数のサブスクリバに ACL を適用します。
  - ステップ5 Exec モードの save configuration コマンドを使用して、フラッシュメモリ、外部メモリデバイス、および/またはネットワークの場所に設定を保存します。コンフィギュレーションファイルを検証して保存する方法の詳細については、『*System Administration Guide*』および『*Command Line Interface Reference*』を参照してください。

この項の設定例で使用されているコマンドは、最もよく使用されているコマンドまたはその可能性の高いコマンド、および/またはキーワードオプションが提示される範囲で、基本機能を提供します。多くの場合は、他のオプションのコマンドやキーワードオプションを使用できます。すべてのコマンドの詳細については、『*Command Line Interface Reference*』を参照してください。すべてのコマンドまたはキーワード/変数を、サポートまたは使用できるわけではありません。可用性はプラットフォームのタイプとインストールされているライセンスによって異なります。

---

## 内部 CSS の IP アクセスリストの定義

IP ACL は、サブスクリバトラフィックのタイプと、トラフィックをリダイレクトする方向 (アップリンク、ダウンリンク、または両方) を指定します。IP ACL は、サブスクリバの認証が実行されるコンテキストで指定する必要があります。



**注意** データ損失のリスクを最小限に抑えるために、システムがサブスクリバセッションを促進しながら、ACL の設定を変更しないようにしてください。

内部 CSS の IP ACL を定義するには、次の設定例を使用します。CLI の Exec モードで開始します。

```
configure
context context_name
ip access-list acl_name
```

```
redirect css service service_name keywords options
end
```

注:

- *service\_name* は、ACL サービス名である必要があります。
- **redirect css service** コマンドで使用可能なキーワードとオプションについては、『*Command Line Interface Reference*』の「*ACL Configuration Mode Commands*」の章を参照してください。
- IPv6 ACL の場合は、IPv6 ACL コンフィギュレーションモードで同じ設定を行う必要があります。『*Command Line Interface Reference*』の「*IPv6 ACL Configuration Mode Commands*」の章を参照してください。

## 個々のサブスクリイバへの ACL の適用 (オプション)

個々のサブスクリイバへの ACL の適用方法の詳細については、「アクセスコントロールリスト」の章の「個々のサブスクリイバへの ACL の適用」の項を参照してください。

## 複数のサブスクリイバへの ACL の適用 (オプション)

IP ACL は、プロファイル内の属性を介してサブスクリイバに適用されます。サブスクリイバプロファイルは、システム上でローカルで設定することも、RADIUS サーバー上でリモートで設定することもできます。

システムには、特定の属性が個々のサブスクリイバのプロファイルに含まれていない場合にデフォルト値として機能するサブスクリイバ機能の設定が用意されています。適切に設定されている場合は、次の機能を使用して ACL を適用できます。

- *default* というサブスクリイバのプロファイルに ACL を適用することで、特定のコンテキスト内で促進されたすべてのサブスクリイバ。
- サブスクリイバのプロファイルに ACL を適用した後、**default subscriber** コマンドを使用してそのサブスクリイバを「デフォルト」のプロファイルとして使用するように設定することで促進されたすべてのサブスクリイバ。

## サブスクリイバの名前付きデフォルトへの ACL の適用 (オプション)

ACL をデフォルトのサブスクリイバに適用する方法については、「アクセスコントロールリスト」の章の「サブスクリイバの名前付きデフォルトへの ACL の適用」の項を参照してください。

## サービス指定のデフォルトのサブスクリイバへの ACL の適用 (オプション)

さまざまなシステムサービスによって「デフォルト」プロファイルとして使用されるように ACL をサブスクリイバに適用する方法については、「アクセスコントロールリスト」の章の「サービス指定のデフォルトのサブスクリイバへの ACL の適用」の項を参照してください。

## APN を介した複数のサブスライバへの ACL の適用（オプション）

IP ACL は、プロファイル内の属性を介してサブスライバに適用されます。サブスライバプロファイルは、システム上でローカルで設定することも、RADIUS サーバー上でリモートで設定することもできます。

設定時間を短縮するために、代わりに ACL を APN テンプレートに適用することができます。設定されている場合、APN テンプレートによって促進されたサブスライバパッケージには、関連付けられた ACL が適用されます。

APN を介して複数のサブスライバに ACL を適用する方法については、「アクセスコントロールリスト」の章の「APN を介した複数のサブスライバへの単一 ACL の適用」の項を参照してください。



## 第 26 章

# セッションリカバリ

堅牢なハードウェアフェールオーバーと冗長性保護により、システム上のハードウェアやソフトウェアの障害を迅速に修正できます。ただし、さまざまな理由でソフトウェアの障害が発生する可能性があり、ほとんどの場合、何の前触れ也没有ありません。

この章では、ハードウェアまたはソフトウェアに障害が発生した場合に、サブスクライバセッション情報のシームレスなフェールオーバーと再構築を行うセッションリカバリ機能について説明します。



**重要** セッションリカバリは、シスコのライセンス対象機能です。別の機能ライセンスが必要になる場合があります。特定のライセンス要件の詳細については、シスコのアカウント担当者にお問い合わせください。ライセンスのインストールと確認の詳細については、「ソフトウェア管理操作」の「ライセンスキーの管理」の項を参照してください。

この章は、次の項で構成されています。

- [セッションリカバリの仕組み \(367 ページ\)](#)
- [セッションリカバリをサポートするためのシステムの設定 \(370 ページ\)](#)
- [制御タスクの統計情報のリカバリ \(374 ページ\)](#)

## セッションリカバリの仕組み

この項では、この機能の実装方法とリカバリプロセスについての概要を示します。

セッションリカバリ機能は、システム内のハードウェアまたはソフトウェアに障害が発生した場合に、サブスクライバセッション情報のシームレスなフェールオーバーと再構築を行い、完全に接続されたユーザーセッションが切断されるのを防ぎます。

セッションリカバリは、システム内の重要なソフトウェアプロセス（セッションマネージャや AAA マネージャなど）をミラーリングすることによって実行されます。これらのミラー化されたプロセスはアイドル状態（スタンバイモード）を維持し、ソフトウェア障害が発生した場合（セッションマネージャのタスクが中止された場合など）に必要なまでいかなる処理も実行しません。

セッションリカバリが適切に動作しない場合があります。これには次が含まれます。

- セッションリカバリ動作中に、新たなソフトウェアまたはハードウェアの障害が発生した場合。たとえば、AAA マネージャは、そこに含まれていた状態情報が、新たにアクティブになったセッションマネージャのタスクの入力に使用されている間に失敗します。
- セッションリカバリをサポートするためのハードウェアのリソース（パケット処理カードのメモリや制御プロセッサ）が不足している場合。



**重要** セッションリカバリ動作の後、マネージャごと（AAA マネージャ、セッションマネージャなど）に収集および保持された統計情報など、一部の統計情報は、通常は回復されません。アカウントリングおよび課金関連の情報のみがチェックポイントリングされ、回復します。

セッションリカバリは、次の機能で使用できます。

- L2TP LAC サポートを必要とするセッション（HA や GGSN セッションの上で再生成された PPP を除く）
- ASR 5500 のみ：簡易 IP、モバイル IP、およびプロキシモバイル IP をサポートしているクローズド RP PDSN サービス
- ASR 5500 のみ：eHRPD サービス（進化した高レートパケットデータ）
- ASR 5500 のみ：ePDG サービス（進化したパケットデータゲートウェイ）
- IPv4 コンテキストと PPP PDP コンテキストの GGSN サービス
- ユーザーごとのレイヤ 3 トンネルの有無にかかわらず、モバイル IP および/またはプロキシモバイル IP セッションタイプをサポートしている HA サービス
- ASR 5500 のみ：HNB-GW：IuH を介した HNB セッション
- ASR 5500 のみ：HNB-GW：IuPS と IuCS を介した HNB-CN セッション
- ASR 5500 のみ：HNB-GW：SeGW セッションの IPSec トンネル
- ASR 5500 のみ：IPv4 用の HSGW サービス
- IPCF（インテリジェントポリシー制御機能）
- ASR 5500 のみ：IPSG 専用システム（IP サービスゲートウェイ）
- LNS セッションタイプ（L2TP ネットワークサーバー）
- MME（モビリティ マネージメント エンティティ）
- ASR 5500 のみ：NEMO（ネットワークモビリティ）
- IPv4 用の P-GW サービス
- ASR 5500 のみ：PDIF（パケット データ インターワーキング機能）
- 簡易 IP、モバイル IP、およびプロキシモバイル IP をサポートしている PDSN サービス



- S-GW (サービングゲートウェイ)
- SGSN (サービング GPRS サポートノード) サービス
- ASR 5000 と VPC-DI : IPv6 と IPv4IPv6 (デュアル) の PDP セッションリカバリは、3G サービスと 2G サービスでサポートされます。
- SaMOG (GTP を介した S2a モビリティ) ゲートウェイ (CGW と MRME)
- ASR 5500 のみ : SAE-GW (System Architecture Evolution ゲートウェイ)
- ASR 5500 のみ : IPv4 コンテキストと PPP PDP コンテキスト用の SGSN サービス (3G サービスと 2.5 G サービス)

セッションリカバリは、次の機能ではサポートされていません。

- 接続先ベースのアカウントイングリカバリ
- GGSN ネットワークによって開始された接続
- 2 つ以上のサービスインスタンスを使用する GGSN セッション
- IPSec 統合を使用した MIP/L2TP
- 複数の同時バインディングを使用した MIP セッション
- L2TP を使用したモバイル IP セッション
- 複数の MIP セッション
- :RAB リカバリ



---

**重要** その他の可能なセッションリカバリおよびシャード間セッションリカバリ (ICSR) のサポートの制限については、常に個々の製品のアドミニストレーションガイドを参照してください。

---

セッションリカバリが発生すると、システムは次のサブスクリバ情報を再構築します。

- 正しいコール動作を維持するために必要なデータと制御状態情報。
- サブスクリバデータの最小の統計情報セット。アカウント情報を実際に保持するために必要です。
- コール期間、絶対時間などのさまざまなタイマー値を回復するためのベストエフォート型の試行。
- アイドル時間タイマーをゼロにリセットし、再登録タイマーを HA セッションの最大値にリセットして、セッションリカバリに対してより控えめなアプローチをとります。



---

**重要** 部分的に接続されたコール (HA 認証が保留されていても、AAA サーバーによってまだ応答確認されていないセッションなど) は、障害が発生しても回復されません。

---



- (注) クリティカルなタスクが失敗すると、StarOSが再起動されます。カーネルの障害、ハイパーバイザの障害、またはハードウェアの障害によって、VMが再起動するか、またはオフラインになります。これらのタイプの障害の解決策として、2つのVPC-DI間または2つのVPC-SI間でICSRを使用することを推奨します。

## セッションリカバリをサポートするためのシステムの設定

次に、現在、動作している運用システム（着信コールを受け入れることができる）か、動作していないシステム（実稼働ネットワークに含まれておらず、そのためライブサブスクライバ/カスタマーのデータを処理していない）のいずれかに対してセッションリカバリ機能を設定する手順を示します。



- 重要** 機能使用キーがある場合でも、デフォルトでは、セッションリカバリ機能はシステム上で無効になっています。

## セッションリカバリの有効化

前述したように、アウトオブサービス（OOS）でコンテキストがまだ設定されていないシステム、または現在コールを処理できるインサービスシステムで、セッションリカバリを有効にすることができます。ただし、システムがインサービスである場合は、セッションリカバリ機能が有効になる前に再起動する必要があります。

### アウトオブサービスシステムでのセッションリカバリの有効化

次に、コンテキストが設定されていないシステムの場合の手順を示します。

アウトオブサービスシステムでセッションリカバリ機能を有効にするには、次の手順を実行します。この手順は、Execモードプロンプトで開始することを前提としています。

- ステップ1** Execモードのプロンプトで、**show license info** コマンドを実行し、システム上のセッションと機能の使用ライセンスを介してセッションリカバリ機能が有効になっていることを確認します。

セッションリカバリ機能の現在のステータスが [Disabled] になっている場合は、ライセンスキーをシステムにインストールするまで、この機能を有効にすることはできません。

- ステップ2** セッションリカバリを有効にするには、次の設定例を使用します。

```
configure
require session recovery
end
```

(注) このコマンドを設定した後で、設定を保存してからシャーシをリロードし、コマンドを有効にする必要があります。設定ファイルを保存してシャーシをリロードする方法については、使用している展開の『System Administration Guide』を参照してください。

**ステップ3** 「設定の確認と保存」の説明に従って、設定を保存します。

起動すると、システムはセッションリカバリを有効にし、ミラーリングされた「スタンバイモード」のタスクをすべて作成し、パケット処理カードの予約やその他の操作を自動的に実行します。

**ステップ4** システムが設定され、インサービスになったら、この機能をサポートするための準備がシステムに整っていることを管理します。次の説明にしたがってください。 [セッションリカバリステータスの表示 \(372 ページ\)](#)

---

## インサービスシステムでのセッションリカバリの有効化

すでに保存されている設定があるシステムでセッションリカバリを有効にすると、セッションリカバリコマンドは、設定ファイル内のサービス コンフィギュレーション コマンドの前に自動的に配置されます。

インサービスシステムでセッションリカバリ機能を有効にするには、次の手順に従います。この手順は、Exec モードプロンプトで開始することを前提としています。

---

**ステップ1** Exec モードプロンプトで **show license info** コマンドを実行して、セッションと機能の使用ライセンスを使用して、セッションリカバリ機能が有効になっていることを確認します。

セッションリカバリ機能の現在のステータスが [Disabled] になっている場合は、ライセンスキーをシステムにインストールするまで、この機能を有効にすることはできません。

**ステップ2** セッションリカバリを有効にするには、次の設定例を使用します。

```
configure
require session recovery
end
```

この機能は、システムが再起動されるまでは有効になりません。

**ステップ3** 「設定の確認と保存」の説明に従って、設定を保存します。

**ステップ4** **reload** コマンドを入力して、システムの再起動を実行します。

次のプロンプトが表示されます。

Are you sure? [Yes|No]:

**Yes** と入力して、システム再起動の実行を確認します。

再起動すると、システムはセッションリカバリを有効にし、ミラーリングされた「スタンバイモード」のタスクをすべて作成し、パケット処理カードの予約やその他の操作を自動的に実行します。

**ステップ5** システムが再起動したら、システムがこの機能をサポートするように準備されていることを確認する必要があります。次の説明に従ってください。[セッションリカバリ ステータスの表示 \(372 ページ\)](#)

上級ユーザーはテキストユーザーなどの手段を使用して、**require session recovery** コマンドシンタックスを既存の設定ファイルに挿入してから、設定ファイルを手動で適用することもできます。このコマンドが既存の設定ファイルの最初の数行の間に配置されるようにする際は注意してください。これは、ローカル以外のコンテキストの作成よりも前に表示される必要があります。

## セッションリカバリ機能の無効化

システムでセッションリカバリ機能を無効にするには、グローバル コンフィギュレーション モードのプロンプトから **no require session recovery** コマンドを入力します。



**重要** このコマンドがインサービシステムで発行された場合は、**reload** コマンドを発行してシステムを再起動する必要があります。

## セッションリカバリ ステータスの表示

システムがセッションリカバリを実行できるかどうかを確認するには、イネーブルにして、Exec モードプロンプトから **show session recovery status verbose** コマンドを入力します。

このコマンドの出力は、次に示す例のようになります。

```
[local]host_name# show session recovery status
Session Recovery Status:
  Overall Status           : SESSMGR Not Ready For Recovery
  Last Status Update      : 1 second ago

[local]host_name# show session recovery status
Session Recovery Status:
  Overall Status           : Ready For Recovery
  Last Status Update      : 8 seconds ago

[local]host_name# show session recovery status verbose
Session Recovery Status:
  Overall Status           : Ready For Recovery
  Last Status Update      : 2 seconds ago
```

cpu state	----sessmgr----		----aaamgr----		demux		status
	active	standby	active	standby	active	status	
1/1 Active	2	1	1	1	0	Good	
1/2 Active	1	1	0	0	0	Good	
1/3 Active	1	1	3	1	0	Good	
2/1 Active	1	1	1	1	0	Good	
2/2 Active	1	1	0	0	0	Good	
2/3 Active	2	1	3	1	0	Good	
3/0 Active	0	0	0	0	1	Good (Demux)	
3/2 Active	0	0	0	0	1	Good (Demux)	
4/1 Standby	0	2	0	1	0	Good	

```

4/2 Standby 0 1 0 0 0 Good
4/3 Standby 0 2 0 3 0 Good
[local]host_name#

```

## セッション情報回復の確認

セッション状態情報とセッションリカバリステータスを表示するには、次のコマンドを入力します。

```
[local]host_name# show subscriber debug-info { callid id | msid id | username name }
```

次に、このコマンドのセッションリカバリ操作が実行される前と後の両方の出力例を示します。わかりやすくするために、この例の [Redundancy Status] フィールドを太字で示しています。

```

username: user1          callid: 01call1b1          msid: 0000100003
Card/Cpu: 4/2
Sessmgr Instance: 7
Primary callline:
Redundancy Status: Original Session
  Checkpoints      Attempts      Success      Last-Attempt      Last-Success
  Full:            69           68           29800ms          29800ms
  Micro:           206          206          20100ms          20100ms
Current state: SMGR_STATE_CONNECTED
FSM Event trace:
  State              Event
  SMGR_STATE_OPEN    SMGR_EVT_NEWCALL
  SMGR_STATE_NEWCALL_ARRIVED SMGR_EVT_ANSWER_CALL
  SMGR_STATE_NEWCALL_ANSWERED SMGR_EVT_LINE_CONNECTED
  SMGR_STATE_LINE_CONNECTED SMGR_EVT_LINK_CONTROL_UP
  SMGR_STATE_LINE_CONNECTED SMGR_EVT_AUTH_REQ
  SMGR_STATE_LINE_CONNECTED SMGR_EVT_IPADDR_ALLOC_SUCCESS
  SMGR_STATE_LINE_CONNECTED SMGR_EVT_AUTH_SUCCESS
  SMGR_STATE_LINE_CONNECTED SMGR_EVT_UPDATE_SESS_CONFIG
  SMGR_STATE_LINE_CONNECTED SMGR_EVT_LOWER_LAYER_UP

Data Reorder statistics
Total timer expiry: 0          Total flush (tmr expiry): 0
Total no buffers: 0          Total flush (no buffers): 0
Total flush (queue full): 0  Total flush (out of range): 0
Total flush (svc change): 0  Total out-of-seq pkt drop: 0
Total out-of-seq arrived: 0

IPv4 Reassembly Statistics:
  Success: 0          In Progress: 0
  Failure (timeout): 0  Failure (no buffers): 0
  Failure (other reasons): 0

Redirected Session Entries:      Allowed:
2000      Current: 0
          Added: 0          Deleted:
          0
          Revoked for use by different subscriber: 0
Peer callline:
Redundancy Status: Recovered Session
  Checkpoints      Attempts      Success      Last-Attempt      Last-Success
  Full:            0            0            0ms              0ms
  Micro:           0            0            0ms              0ms
Current state: SMGR_STATE_CONNECTED
FSM Event trace:
  State              Event
  SMGR_STATE_LINE_CONNECTED SMGR_EVT_LOWER_LAYER_UP
  SMGR_STATE_CONNECTED    SMGR_EVT_AUTH_REQ

```

```

SMGR_STATE_CONNECTED          SMGR_EVT_AUTH_SUCCESS
SMGR_STATE_CONNECTED          SMGR_EVT_REQ_SUB_SESSION
SMGR_STATE_CONNECTED          SMGR_EVT_RSP_SUB_SESSION
SMGR_STATE_CONNECTED          SMGR_EVT_ADD_SUB_SESSION
SMGR_STATE_CONNECTED          SMGR_EVT_AUTH_REQ
SMGR_STATE_CONNECTED          SMGR_EVT_AUTH_SUCCESS
SMGR_STATE_CONNECTED          SMGR_EVT_AUTH_REQ
SMGR_STATE_CONNECTED          SMGR_EVT_AUTH_SUCCESS
SMGR_STATE_CONNECTED          SMGR_EVT_AUTH_REQ
SMGR_STATE_CONNECTED          SMGR_EVT_AUTH_SUCCESS
SMGR_STATE_CONNECTED          SMGR_EVT_AUTH_REQ
SMGR_STATE_CONNECTED          SMGR_EVT_AUTH_SUCCESS
SMGR_STATE_CONNECTED          SMGR_EVT_AUTH_REQ
SMGR_STATE_CONNECTED          SMGR_EVT_AUTH_SUCCESS
SMGR_STATE_CONNECTED          SMGR_EVT_AUTH_REQ
SMGR_STATE_CONNECTED          SMGR_EVT_AUTH_SUCCESS
Data Reorder statistics
  Total timer expiry:          0          Total flush (tmr expiry): 0
  Total no buffers:            0          Total flush (no buffers): 0
  Total flush (queue full):    0          Total flush (out of range):0
  Total flush (svc change):    0          Total out-of-seq pkt drop: 0
  Total out-of-seq arrived:    0
IPv4 Reassembly Statistics:
  Success:                     0          In Progress:                0
  Failure (timeout):           0          Failure (no buffers):       0
  Failure (other reasons):     0
Redirected Session Entries:
  Allowed:                     2000        Current:                    0
  Added:                       0          Deleted:                    0
  Revoked for use by different subscriber: 0

```

## 制御タスクの統計情報のリカバリ

リカバリ制御タスク（RCT）の統計情報には、次の情報が表示されます。

- リカバリアクションの実行：移行、シャットダウン、スイッチオーバー
- イベントのタイプ：計画済みまたは未計画
- カードからカードへ：スロット番号
- 開始時刻：YYYY-MMM-DD+hh:mm:sss.sss
- 期間：秒
- カード障害のデバイス（CPU $n$  など）
- カード障害の理由
- カードが使用可能な状態であるか、障害はない
- リカバリアクションのステータス：成功または失敗の理由
- リカバリアクションが失敗した場合、失敗したタイムスタンプ
- リカバリアクションが失敗した場合、失敗したタスクファシリティ名
- リカバリアクションが失敗した場合、失敗したインスタンス番号

## show rct stats コマンド

Exec モードの **show rct stats** コマンドでは、次のシンタックスが採用されています。

```
[local]host_name# show rct stats [verbose]
```

`verbose` キーワードを指定しないと、次の例に示すようにサマリー出力が表示されます。

```
RCT stats details (Last 1 Actions)

# Action          Type      From To Start Time                Duration      Status
-----
1 Migration(st) Planned    2  1 2016-Jul-12+13:12:21.865    0.003 sec    Success

RCT stats summary
-----
Migrations =                0
Management Card:    0          Average time: 0.000 sec
Packet Card   :    1          Average time: 0.006 sec
Switchovers =                1, Average time - 25.855 sec
```

`verbose` キーワードを指定すると、[show rct stats verbose の出力例 \(375 ページ\)](#) に示す詳細な統計情報が提供されます。

## show rct stats verbose の出力例

```
[local]host_name# show rct stats verbose

RCT stats Details (Last 5 Actions)

Stats 1:
Action          : Migration
Type            : Planned
From            : 5
To              : 6
Start Time      : 2017-Apr-04+03:02:00.132
Failure Reason  : CPU_CRITICAL_TASK_FAILURE
Failure Device  : CPU_0
Is Card Usable  : Yes
Recovery Status: Success
Facility        : N.A
Instance        : N.A
Duration        : 066.050 sec
Graceful        : Enabled
Recovered [1]   : [f:sessmgr, i:6, cpu:50, pid:13170]
Recovered [2]   : [f:sessmgr, i:3, cpu:50, pid:13167]

RCT stats Details (Last 5 Actions)

Stats 2:
Action          : Shutdown
From            : 12
To              : 13
Start Time      : 2017-Apr-04+03:02:10.100
Is Card Usable  : Yes
Failure Reason  : NPU_LC_CONNECT_TOP_FAIL
Failure Device  : PAC_LC_CONNECT_HARDWARE
Recovery Status: Success
Facility        : N.A
Instance        : N.A
Duration        : 002.901 sec
Graceful        : Enabled
Recovered [1]   : [f:sessmgr, i:6, cpu:50, pid:13170]
Recovered [2]   : [f:sessmgr, i:3, cpu:50, pid:13167]

Stats 3:
Action          : Migration
```

## show rct stats verbose の出力例

```

From          : 7
To            : 11
Start Time    : 2017-Apr-04+03:03:40.120
Is Card Usable : Yes
Failure Reason : N.A.
Failure Device : N.A
Recovery Status : Success
Facility      : N.A
Instance      : N.A
Duration      : 003.423 sec
Graceful      : Enabled
  Recovered [1] : [f:sessmgr, i:6, cpu:50, pid:13170]
  Recovered [2] : [f:sessmgr, i:3, cpu:50, pid:13167]

Stats 4:
Action        : Migration
From          : 7
To            : 11
Start Time    : 2017-Apr-04+03:03:41.256
Is Card Usable : Yes
Failure Reason : N.A.
Failure Device : N.A
Recovery Status : TASK_MIGRATION_FAIL_PREMIGRATE
Facility      : vpnmgr
Instance      : 13
Duration      : 005.222 sec
Graceful      : Enabled
  Recovered [1] : [f:sessmgr, i:6, cpu:50, pid:13170]
  Recovered [2] : [f:sessmgr, i:3, cpu:50, pid:13167]

Stats 5:
Action        : Migration
From          : 6
To            : 7
Start Time    : 2017-Apr-04+04:18:30.106
Is Card Usable : Yes
Failure Reason : N.A.
Failure Device : N.A
Recovery Status : TASK_MIGRATION_FAIL_RENAME
Facility      : sessmgr
Instance      : 63
Duration      : 004.134 sec
Graceful      : Enabled
  Recovered [1] : [f:sessmgr, i:6, cpu:50, pid:13170]
  Recovered [2] : [f:sessmgr, i:3, cpu:50, pid:13167]

RCT stats Summary
-----
Migrations = 3, Average time = 4.260 sec
Switchovers = 0

```





## 第 27 章

# シャーシ間セッションリカバリ

この章では、シャーシ間セッションリカバリ（ICSR）を設定する方法について説明します。製品アドミニストレーションガイドには、システム上での基本サービスの設定例と手順が示されています。以下で説明する手順を実行する前に、それぞれの製品管理ガイドの説明に従って、サービスモデルに最適な設定例を選択し、そのモデルに必要な要素を設定する必要があります。

VPC-SI のコンテキストでは、シャーシはハイパーバイザ内で VPC-SI を実行するように設定されたサーバーです。ICSR は、同様に設定された別のサーバーで実行されている VPC-SI インスタンスに対してフェールオーバー保護を提供します。



**重要** ICSR は、別個のライセンスを必要とするライセンス供与されたシスコの機能です。特定のライセンス要件の詳細については、シスコのアカウント担当者にお問い合わせください。ライセンスのインストールと確認の詳細については、「ソフトウェア管理操作」の「ライセンスキーの管理」の項を参照してください。

この章では、次の項目について説明します。

- [概要](#) (377 ページ)
- [ICSR の運用](#) (383 ページ)
- [ICSR の設定](#) (386 ページ)
- [ICSR 運用のトラブルシューティング](#) (403 ページ)
- [オペレーティングシステムの更新](#) (404 ページ)

## 概要

ICSR 機能は、サブスクリバのサービスを中断することなく、継続的なコール処理を可能にする最も高い可用性を提供します。ICSR により、オペレータは冗長性を確保するためにゲートウェイを設定することができます。ゲートウェイで障害が発生した場合、ICSR はその障害を迂回してセッションを透過的にルーティングできるため、ユーザーエクスペリエンスが維持されます。ICSR では、セッション情報と状態も維持されます。

システムは、同じ StarOS リリースで ICSR をサポートする 2 つのインスタンス間で ICSR をサポートしています。複数のサービスタイプが使用されている VM の組み合わせでは、ICSR をサポートするサービスのみが ICSR を使用できます。

ICSR は、サイト/行/ラック/ホストの停止や主要なソフトウェア障害に冗長性を提供できます。重複しないホストとネットワーク インターコネクト上で 2 つのインスタンスを実行する必要があります。ICSR は、同じように設定されている VPC-DI インスタンスまたは VPC-SI インスタンスの間でのみサポートされます。

VPC-SIL2 と L3 の両方の ICSR をサポートします。

ICSR は冗長仮想シャージを使用して実装されます。各 VPC-SI インスタンスの仮想シャージは、プライマリとバックアップとして設定されます。1 つはアクティブで、もう 1 つはスタンバイです。両方の仮想シャージが同じ AAA サーバーに接続されています。チェックポイントの期間タイマーは、アクティブシャージからスタンバイシャージにサブスクライバデータが送信されるタイミングを制御します。コールトラフィックを処理しているアクティブシャージがアウトオブサービスになると、スタンバイシャージはアクティブ状態に移行し、サブスクライバセッションを中断せずにコールトラフィックの処理を続行します。

仮想シャージは、サービス冗長性プロトコル (SRP) リンクと呼ばれる独自の TCP ベースの接続を介して、どちらがアクティブであるかを判別します。SRP リンクは、プライマリシャージとバックアップシャージのアクティブな CF 間で Hello メッセージを交換するために使用され、適切なシステム運用のために維持する必要があります。詳細については、「セッションリカバリ」の章を参照してください。

ICSR ライセンスは、次のサービスに対して現在サポートされています。

- GGSN ゲートウェイ GPRS サポートノード
- P-GW Packet Data Network Gateway
- S-GW サービングゲートウェイ
- SAE-GW System Architecture Evolution ゲートウェイ

ICSR の L2TP アクセスコンセントレータ (LAC) 機能は、次のプロトコルおよびサービスでサポートされています。

- eGTP 拡張 GPRS トンネリングプロトコル
- GGSN ゲートウェイ GPRS サポートノード
- P-GW Packet Data Network Gateway
- SAEGW System Architecture Evolution ゲートウェイ

ICSR の L2TP アクセスコンセントレータ (LAC) 機能は、次のサービスでサポートされていません。

- PMIP プロキシモバイル IP

ICSR の L2TP ネットワークサーバー (LNS) 機能は、どのサービスでもサポートされていません。



(注) LAC の ICSR サポートには、個別の LAC のライセンスとシャーシ間セッションリカバリのライセンスが必要です。



(注) 特定のサービスが ICSR をオプションとしてサポートしているかどうかを確認するには、シスコのアカウント担当者にお問い合わせください。

## シャーシ間の通信

SRP リンクがアウトオブサービスになる状況では、どのシャーシがセッションを処理するかを決定するために優先順位スキームが使用されます。次の優先順位スキームが使用されます。

- route modifier
- chassis priority
- MIO/UMIO/MIO2 MAC address

## チェックポイントメッセージ

詳細については、付録の「ICSR のチェックポイントイング」を参照してください。

## SRP CLI コマンド

### Exec モード CLI コマンド

Exec モードの **srp** CLI 設定コマンドを使用して、SRP 機能を有効または無効にしたり、開始することができます。次の表に、これらのコマンドの一覧と簡単な説明を示します。詳細については、『*Command Line Interface Reference*』の「*Exec Mode Commands (D-S)*」の章を参照してください。

表 27: *srp* CLI コマンド

コマンド	説明
<b>srp disable nack micro-chkpt-cmd</b>	アクティブシャーシから完全なチェックポイントをトリガーする可能性があるスタンバイシャーシからの NACK メッセージの送を無効にします。完全なチェックポイントを送信すると、SRP 帯域幅が増加します。このコマンドは、継続的に障害が発生している特定のマイクロチェックポイントの NACK 機能を無効にします。

コマンド	説明
<b>srp initiate-audit manual-with-sync</b>	ICSR シャーシ間の強制監査を開始します。この監査により、2つの ICSR ピアが同期され、スケジュール済みまたは予定外のスイッチオーバーイベントの前に不一致が識別されます。
<b>srp initiate-switchover</b>	アクティブから非アクティブへの強制スイッチオーバーを実行します。このコマンドをアクティブシャーシで実行した場合、アクティブシャーシを非アクティブ状態、非アクティブシャーシをアクティブ状態に切り替えます。下記の注を参照してください。
<b>srp reset-auth-probe-fail</b>	認証プローブモニターの障害情報を 0 にリセットします。
<b>srp reset-diameter-fail</b>	Diameter モニターの障害情報を 0 にリセットします。
<b>srp terminate-post-process</b>	スイッチオーバー後の処理を強制的に終了します。
<b>srp validate-configuration</b>	アクティブシャーシの設定を検証します。
<b>srp validate-switchover</b>	予定済みの SRP スイッチオーバーの準備がアクティブシャーシとスタンバイシャーシの両方で整っていることを検証します。



**重要** ICSR は手動スイッチオーバーを許可する前に、両方のシャーシでセッションマネージャの接続を確認します。アクティブシャーシ内の1つ以上のセッションマネージャがスタンバイシャーシに接続されていない場合は、スイッチオーバーは開始されません。セッションマネージャの数が一致していないことを通知するメッセージが画面に表示されます。**force** キーワードを使用すると、不一致があってもスイッチオーバーを開始できます。**show checkpoint statistics verbose** コマンドの出力では、ピアシャーシに接続されていないインスタンスについては「peer conn」列のセッションマネージャインスタンス（「smgr inst」）に「Ready」が示されません。

## show コマンド

Exec モードの **show srp** コマンドは、SRP 機能に関連するさまざまな情報を表示します。次の表に、これらのコマンドの一覧と簡単な説明を示します。これらのコマンドの詳細については、『*Command Line Interface Reference*』の「*Exec Mode show Commands (Q ~ S)*」の章を参照してください。

表 28: **show srp** コマンド

コマンド	説明
<b>show srp audit-statistics</b>	外部監査の統計情報を表示します。
<b>show srp call-loss statistics</b>	スイッチオーバー中に失われたコールの履歴を表示します。
<b>show srp checkpoint statistics</b>	セッションの冗長性データ（セッションマネージャ、現在のコールリカバリレコードなど）の統計情報のチェックポイントを表示します。

コマンド	説明
<b>show srp info</b>	サービス冗長プロトコル情報（コンテキスト、シャージの状態、ア、接続状態など）を表示します。
<b>show srp monitor</b>	SRP モニター情報を表示します。
<b>show srp statistics</b>	SRP 統計情報（送信された hello メッセージ、設定の検証、リンクスメッセージ、スイッチオーバーなど）を表示します。

**show srp** コマンドの出力の詳細については、*Statistics and Counters Reference* を参照してください。

## AAA モニター

AAA サーバーは、認証プローブメカニズムを使用してモニターされます。認証プローブが有効な応答を受信した場合、AAA サーバーは稼働していると見なされます。AAA サーバーの設定で指定された **max-retries count** に達した場合、AAA サーバーは停止していると思なされます。設定済みのAAAサーバーのいずれも認証プローブに応答しない場合、SRPはスイッチオーバーを開始します。AAA プローブは、アクティブシャージでのみ実行されます。



**重要** AAA モニタリング障害によるスイッチオーバーのイベントは、非リバーティブです。

新しくアクティブになったシャージが設定済みのAAAサーバーのモニターに失敗した場合、次のいずれかが発生するまでアクティブシャージのままになります。

- 手動スイッチオーバー。
- 別の非AAA障害イベントによって、システムがスイッチオーバーする。
- CLI コマンドを使用してAAA障害フラグをクリアし、シャージをスタンバイに切り替えられるようにする。

## BGP の連携動作

サービス冗長プロトコルは、アドバタイズされたループバック/IP プールルートのルート修飾子値を調整するメカニズムを介して、復元可能なスイッチオーバー動作を実装します。ルート修飾子値の初期値は、設定されたシャージのロールによって決定され、通常の動作値よりも大きい値に初期化されます。これにより、SRP リンク障害やSRP タスク障害が発生した場合でも、ルーティングドメイン内では適切なシャージが優先されるようになります。



**重要** ICSR では、SRP 検証の失敗を回避するために、アクティブシャージとスタンバイシャージで **busyout ip pool** コマンドを同じ順序で設定する必要があります。

アクティブシャーシとスタンバイシャーシは、現在のルート修飾子値を共有します。BGPは、ループバックルートとIPプールルートをアドバタイズするときに、ルート修飾子を自律システム (AS) パスのプリペンドカウントに変換します。アクティブシャーシは常により小さいルート修飾子を持つため、ASパス属性の前により小さい値が付加されます。これにより、ルーティングドメイン内ではこのルートが優先されます。

SRPリンク上の通信が失われ、冗長ペアのシャーシが両方ともアクティブであると主張している場合、以前にアクティブだったシャーシが引き続き優先されます。これは、より小さいASパスをBGPルーティングドメインにアドバタイズしているためです。スイッチオーバーイベントが発生すると、ルート修飾子が増分されます。ロールオーバーを回避するために、ルート修飾子を初期値にリセットするタイミングはしきい値によって決まります。

## 要件

ICSRの設定には、次のものがが必要です。

- 同じサービスタイプに対して同じように設定されたVPC-SIインスタンス。サービスは、SRPによってアクティブ化されたループバックインターフェイスでバインドされている必要があります。両方のインスタンスが同一のハードウェアを備えている必要があります。
- 3つのコンテキスト：
  - **冗長性**：プライマリシャーシとバックアップシャーシの冗長性を設定します。
  - **送信元**：指定されたnas-ip-addressのAAA設定は、HAにバインドされたインターフェイスのIPアドレスであるか、または同じコンテキスト内で設定されたコアネットワークサービスである必要があります。
  - **接続先**：PDNへのモニタリングとルーティングを設定します。
- ボーダーゲートウェイプロトコル (BGP)：ICSRはルート修飾子を使用してシャーシの優先順位を決定します。
- 自律システム (AS) 番号が同じ場合、SRPでアクティブ化されたルートはスタンバイシャーシで自動的に除外されます。それ以外の場合、SRPでアクティブ化されたルートは、ICSRシャーシに接続しているルータでフィルタ処理されるポリシーまたはルートマップによってフィルタ処理されます。
- L3 ICSRは、AS番号が同じ場合に機能するAS-Path属性によってデュアルアクティブシナリオを処理します。
- StarOSはスタンバイのアドバタイズ機能もサポートしています。この機能では、AS-Path属性がスタンバイシャーシのアドバタイズプールを処理します。



**重要** ICSR は、ライセンス供与されたシスコの機能です。これらの手順を使用する前に、各シャージに適切なライセンスがあることを確認します。これを行うには、両方のシャージにログインし、**show license information** コマンドを実行します。「シャージ間セッションリカバリ」を探します。シャージにライセンスがない場合は、シスコのアカウント担当者にお問い合わせください。

スイッチオーバーをトリガーするために、RADIUS プロトコルおよび Diameter のプロトコルをモニターできます。

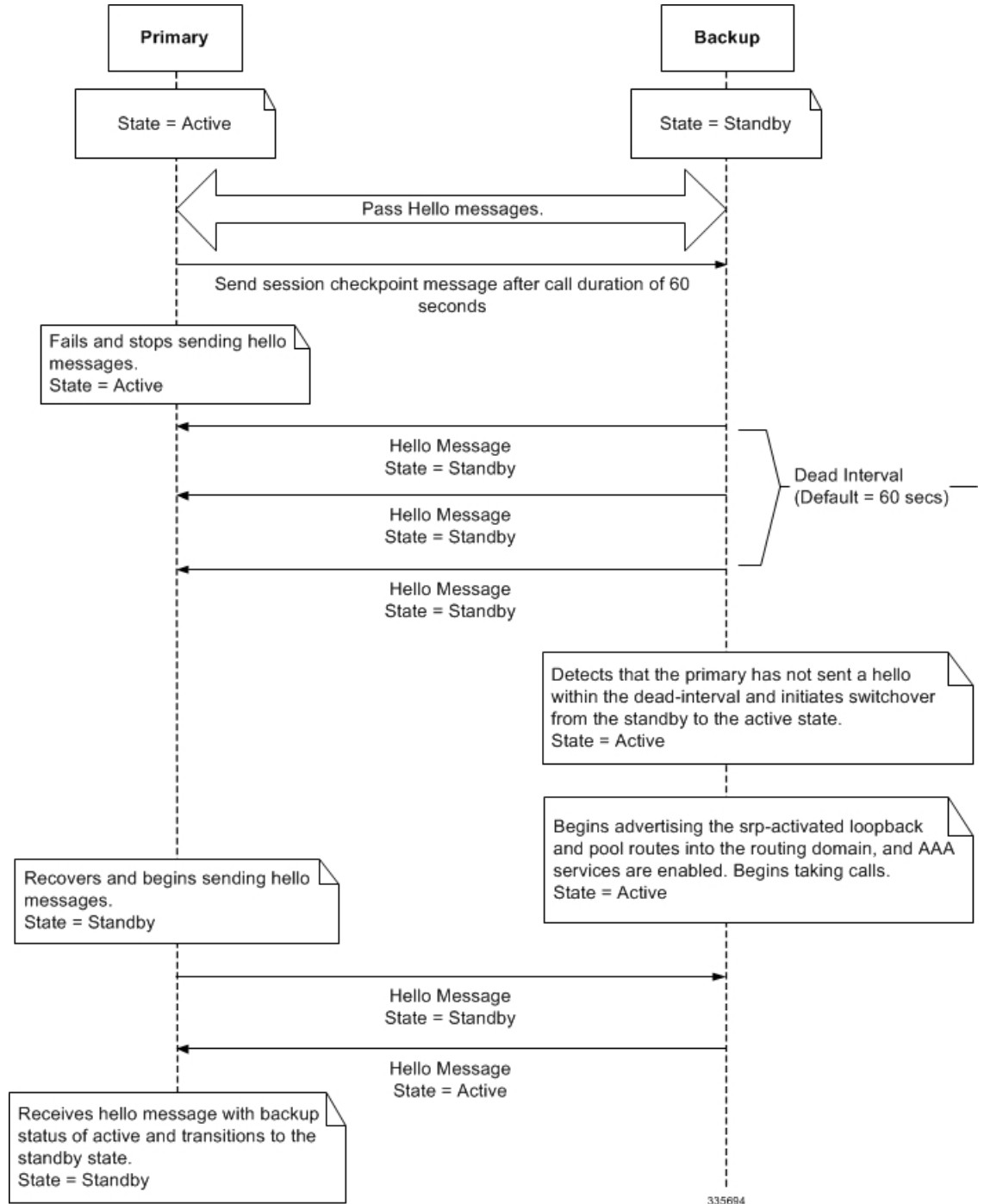
次の図は、ICSR のネットワークを示しています。

## ICSR の運用

このセクションには、ICSR の運用フローが表示されます。

次の図は、プライマリ障害による ICSR プロセスフローを示しています。

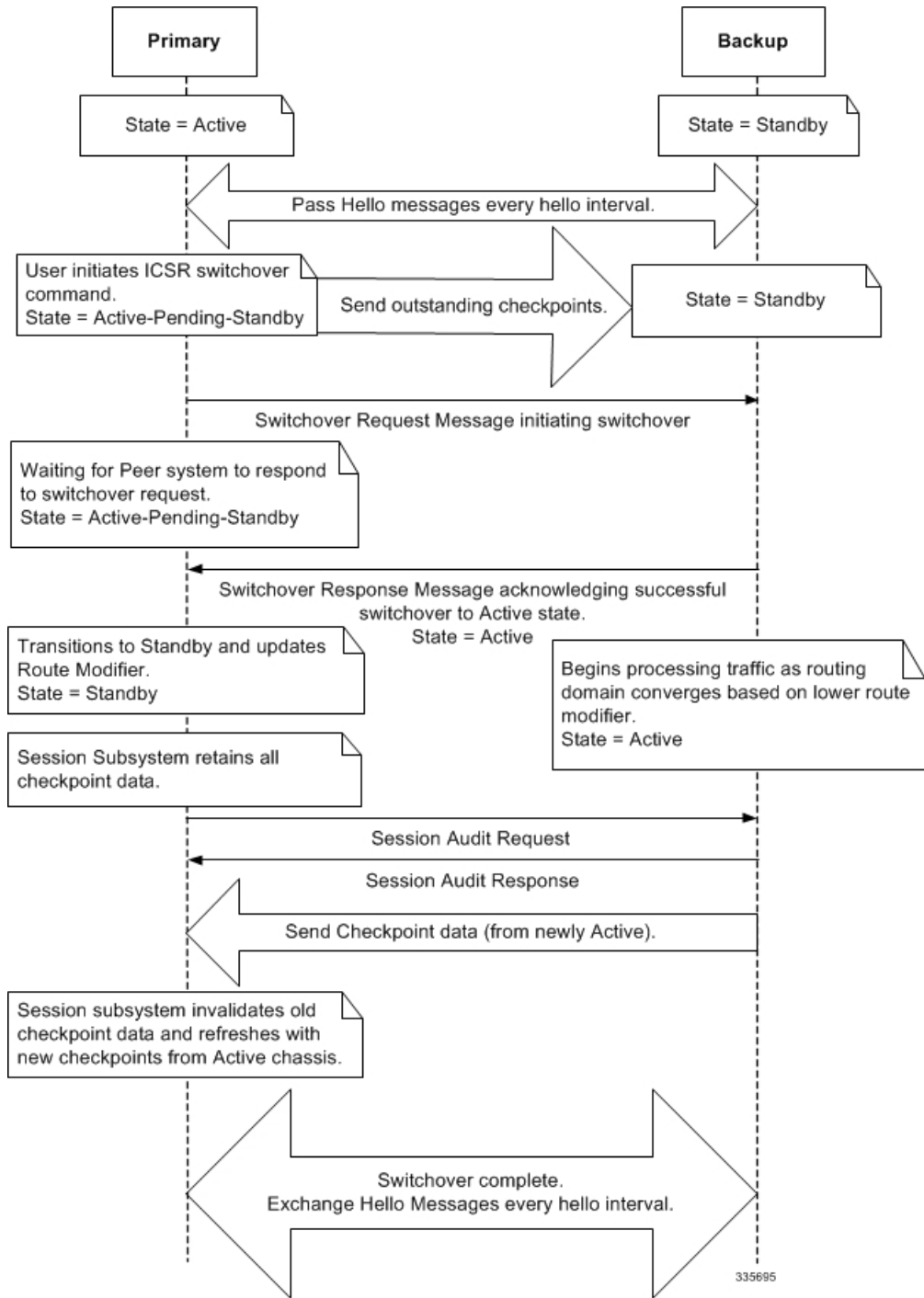
図 9: ICSR プロセスフロー (プライマリ障害)



次の図は、手動スイッチオーバーによる ICSR プロセスフローを示しています。



図 10: ICSR プロセスフロー (手動スイッチオーバー)



335695

## シャーシの初期化

StarOS が各 VPC-SI 仮想シャーシ上で同時に初期化されると、シャーシは設定されたピアに Hello メッセージを送信します。ピアは応答を送信し、シャーシ間の通信を確立し、設定情報を含むメッセージを送信します。

初期化中に、両方の仮想シャーシが同じモードで誤設定されている場合（両方ともアクティブ（プライマリ）または両方ともスタンバイ（バックアップ））、最も高い優先順位を持つシャーシ（ICSR の **priority** コマンドで設定した最も小さい数字）がアクティブになり、もう一方のシャーシはスタンバイになります。

シャーシの優先順位が同じ場合は、StarOS がアクティブ 2 つの MAC アドレスを比較し、高い MAC アドレスを持つシャーシがアクティブになります。たとえば、VPC-SI インスタンスの MAC アドレスが `00-02-43-03-1C-2B` と `00-02-43-03-01-3B` の場合、最後の 3 つのオクテットセット（最初の 3 つのセットはベンダーコード）が比較されます。この例では、`03-1C-2B` と `03-01-3B` が左から右に比較されます。両方の MAC アドレス内の最初のオクテットのペアが同じであるため、次のペアが比較されます。`01` は `1C` よりも低いため、MAC アドレス `00-02-43-03-1C-2B` を持つ VPC-SI 仮想シャーシがアクティブになり、もう一方のシャーシはスタンバイになります。

## シャーシの動作

この項では、シャーシがどのように通信し、サブスクライバセッションを維持し、シャーシのスイッチオーバーを実行するかについて説明します。

### シャーシ通信

### シャーシスイッチオーバー

## ICSR の設定



**重要** ICSR の設定は、プライマリシャーシとバックアップシャーシで同じである必要があります。各シャーシに異なるサービス冗長性プロトコル（SRP）の設定がある場合、セッションリカバリ機能は機能せず、アクティブシャーシがアウトオブサービスになったときにセッションをリカバリできません。

この項では、各シャーシ上の基本的な ICSR の設定方法を説明します。追加のパラメータとオプションを設定するコマンドについては、『*Command Line Interface Reference*』を参照してください。

次に説明する手順は、下記のを前提としています。

- シャーシは、コアネットワークサービスを使用してインストールおよび設定されている。

サービスの設定に関する詳細情報と手順については、それぞれの製品のアドミニストレーションガイドを参照。

- さらに、IP アドレスプールは **SRP** によって **アクティブ化** されている必要がある。
- AAA サーバーがインストールされ、設定されており、両方のシャーシでアクセス可能であること。

AAA サーバーの設定の詳細については、『*AAA Interface Administration and Reference*』を参照。

- BGP ルータがインストールされ、設定されていること。BGP サービスの設定の詳細については、「ルーティング」を参照。

プライマリシャーシやバックアップシャーシに ICSR を設定するには、次のようにします。

- 
- ステップ 1** サービス冗長性プロトコル (SRP) コンテキストの設定 (388 ページ) の設定例を適用して、SRP コンテキストを設定します。
  - ステップ 2** ICSR の送信元コンテキストの変更 (399 ページ) の設定例を適用して、コアネットワークサービスの送信元コンテキストを変更します。
  - ステップ 3** ICSR の接続先コンテキストの変更 (401 ページ) の設定例を適用して、コアネットワークサービスの接続先コンテキストを変更します。
  - ステップ 4** オプション: スタンバイシステムでのバルク統計情報収集の無効化 (402 ページ) の設定例を適用して、スタンバイシステムでバルク統計情報収集を無効にします。
  - ステップ 5** プライマリ設定とバックアップ設定の確認 (402 ページ) の説明に従って、プライマリシャーシおよびバックアップシャーシの設定を確認します。
  - ステップ 6** 「設定の確認と保存」の説明に従って、設定を保存します。
- 

## SRP チェックポイントの設定

### SRP チェックポイントの設定

シャーシ間セッションリカバリ (ICSR) の設定では、アクティブシャーシとスタンバイシャーシの両方で一部の設定が同一である必要があります。Service Redundancy Protocol (SRP) チェックポイントまたはチェックサムは、アクティブシャーシとスタンバイシャーシの設定を検証します。それらが同一であれば、その設定は適切です。設定が同一でない場合、エラーが発生する可能性があります。SRP チェックポイントをサポートするために、BGP ルータ設定の下に VRF 設定が追加されています。

SRP チェックポイントを設定するには、次の設定を使用します。

```
configure
context context_name
service-redundancy-protocol
```

```
[ no ] monitor bgp vrf-srp-validate
end
```

注：

- **vrf-srp-validate** : BGP VRF 設定の SRP 検証を有効にします。
- **no** : BGP VRF 設定の SRP 検証を無効にします。

## モニターリングおよびトラブルシューティング

この項では、この機能をサポートするために `show` コマンドを使用して監視およびトラブルシューティングする方法について説明します。

### show コマンドと出力

この項では、この機能の `show` コマンドとそれらの出力に関する情報を示します。

#### `show configuration srp`

このコマンドの出力が強化され、次のフィールドが表示されるようになりました。

表 29: `show configuration srp` コマンド出力の説明

フィールド	説明
vrf-srp-validate	BGP VRF 設定の SRP 検証を有効にします。

## サービス冗長性プロトコル (SRP) コンテキストの設定

ICSR と連携するようにシステムを設定するには、次のようにします。

- 
- ステップ 1** [SRP コンテキストの作成とバインド \(389 ページ\)](#) の設定例を適用して、シャーシ冗長性コンテキストを作成し、プライマリシャーシの IP アドレスにバインドします。VPC-DI インスタンスの場合、これはプライマリ VPC-DI インスタンス内のアクティブな CF の IP アドレスとなります。
  - ステップ 2** [SRP コンテキストパラメータの設定 \(389 ページ\)](#) の設定例を適用し、優先順位、シャーシモード、hello 間隔、dead 間隔、およびピア IP アドレスを使用してシャーシ冗長性コンテキストを設定します。
  - ステップ 3** [SRP コンテキストのインターフェイスパラメータの設定 \(396 ページ\)](#) の設定例を適用し、シャーシ間通信のインターフェイスパラメータ (インターフェイス名、IP アドレス、およびポート番号を含む) を使用して SRP コンテキストを設定します。
  - ステップ 4** [SRP 設定の確認 \(399 ページ\)](#) の説明に従って、SRP コンテキストの設定を確認します。
  - ステップ 5** 「設定の確認と保存」の説明に従って、設定を保存します。
-

## SRP コンテキストの作成とバインド

次の例を使用して SRP コンテキストを作成し、プライマリシャーシの IP アドレスにバインドします。



**重要** ICSR は 2 つのシャーシ上に設定されます。両方のシステム上で冗長性コンテキストを必ず作成してください。CLI コマンドは両方のシステムで実行する必要があります。続行する前に、両方のシャーシにログインします。常に、プライマリシャーシで設定の変更を最初に行います。この設定を開始する前に、プライマリとして設定するシャーシを特定し、そのログインセッションを使用します。



**重要** ICSR は 2 つの VPC-DI インスタンス上に設定されます。両方のシステム上で冗長性コンテキストを必ず作成してください。CLI コマンドは両方のシステムで実行する必要があります。続行する前に、両方のアクティブ CF にログインします。常に、プライマリ VPC-DI インスタンスのアクティブ CF で設定の変更を最初に行います。この設定を開始する前に、プライマリとして設定する VPC-DI を特定し、そのログインセッションを使用します。

```
configure
context srp_ctxt_name [-noconfirm]
service-redundancy-protocol
  bind address ip_address
end
```

注：

- ICSR は個別のコンテキストで設定および維持する必要があります。
- 必ず、ローカル IP アドレスをプライマリシャーシにバインドしてください。バックアップシャーシを設定するときは、必ずローカル IP アドレスをバックアップシャーシにバインドしてください。

## SRP コンテキストパラメータの設定



**重要** CLI コマンドは、両方の VPC インスタンスで実行する必要があります。続行する前に、両方のアクティブ CF にログインします。最初にプライマリ VPC インスタンスで設定の変更を行います。

### 基本パラメータ

この設定では、シャーシモードと優先順位が割り当てられ、プライマリシャーシとバックアップシャーシ間の冗長性リンクも設定されます。

```

configure
context srp_ctxt_name
  service-redundancy-protocol
    chassis-mode { primary | backup }
    priority priority
    peer-ip-address ip_address
    hello-interval dur_sec
    dead-interval dead_dur_sec
  end

```

注：

- ICSR は個別のコンテキストで設定および維持する必要があります。
- バックアップシャージにシャージモードを割り当てる場合は、**backup** キーワードを必ず入力します。
- **checkpoint** コマンドは、シャージが既存のコールセッションを指していることを確認するまでに待機する時間を設定します。チェックポイントは、IMS (VoLTE) や非 IMS セッションに設定できます。チェックポイントは、障害が発生した場合に実行を再開するために使用できる現在のアプリケーションの状態のスナップショットです。デフォルトの設定は 60 秒です。
- この優先順位によって、両方のシャージが同じシャージモードで誤って設定されている場合に、どのシャージがアクティブになるかが決定されます ([シャージの初期化 \(386 ページ\)](#) を参照)。優先順位の高いシャージの方が小さい番号になります。各シャージに異なる優先順位を割り当てるようにしてください。
- バックアップシャージの IP シャージを **peer-ip-address** として、プライマリシャージに入力します。プライマリシャージの IP アドレスを **peer-ip-address** として、バックアップシャージに割り当てます。
- **dead** 間隔は、**hello** 間隔の 3 倍以上にする必要があります。たとえば、hello 間隔が 10 の場合、dead 間隔は 30 以上にする必要があります。Hello 間隔とデッド間隔が適切に設定されていない場合、システムパフォーマンスに重大な影響が及びます。オプションの **delay-interval** コマンドを使用すると、コンフィギュレーションファイルのロード後のある一定の時間間隔にわたって、デッド間隔の開始を遅延させることができます。

## SRP 冗長性、AAA ガードタイマー、Diameter ガードタイマー

ガードタイマーを使用すると、再起動やタスクの再起動といったローカルの障害が発生しても、ICSR イベントが中断することはありません。

**guard timer** コマンドは、SRP サービスのモニタリングのための **edundancy-guard-period** と **monitor-damping-period** を設定します。

```

configure
context context_name
  service-redundancy-protocol variable
    guard-timer { aaa-switchover-timers { damping-period seconds |
guard-period seconds } | diameter-switchover-timers { damping-period
seconds | guard-period seconds } | srp-redundancy-timers { aaa {

```

```
damping-period seconds | guard-period seconds } | bgp { damping-period
seconds | guard-period seconds } | diam { damping-period seconds |
guard-period seconds } }
end
```

注：

- **aaa-switchover-timers**：ネットワークのコンバージ中に AAA の障害（post ICSR switchover）が発生した場合に、バックツーバックの ICSR スイッチオーバーを防止するタイマーを設定します。
  - **damping-period**：guard-period 内のモニタリング障害による ICSR スイッチオーバーをトリガーする遅延時間を設定します。
  - **guard-period**：local-failure-recovery と network-convergence のタイマーを設定します。
- **diameter-switchover-timers**：ネットワークのコンバージ中に Diameter の障害（post ICSR switchover）が発生した場合に、バックツーバックの ICSR スイッチオーバーを防止するタイマーを設定します。
  - **damping-period**：guard-period 内のモニタリング障害による ICSR スイッチオーバーをトリガーする遅延時間を設定します。
  - **guard-period**：local-failure-recovery と network-convergence のタイマーを設定します。
- **srp-redundancy-timer**：システムがローカルの card-reboot/critical-task-restart の障害からリカバリされている間に、ICSR スイッチオーバーを防止するタイマーを設定します。
  - **aaa**：ローカル障害に続いて AAA のモニタリングの障害が発生する。
  - **bgp**：ローカル障害に続いて BGP のモニタリングの障害が発生する。
  - **diam**：ローカル障害に続いて Diameter のモニタリングの障害が発生する。

## SRP メッセージの DSCP マーキング

SRP 制御およびチェックポイントメッセージの個別の DSCP マーキングを有効にすることができます。**dscp-marking** コマンドは、SRP 制御およびチェックポイント（セッションメンテナンス）メッセージの DSCP マーキング値を設定します。

```
configure
context context_name
  service-redundancy-protocol
    dscp-marking { control | session } dscp_value
```

注：

- **dscp\_value** は次のようになります。
  - **af11**：Assured Forwarding Class 1 Low（ホップごとの動作）
  - **af12**：Assured Forwarding Class 1 Medium ドロップ PHB
  - **af13**：Assured Forwarding Class 1 High ドロップ PHB

- **af21** : Assured Forwarding Class 2 Low ドロップ PHB
- **af22** : Assured Forwarding Class 2 Medium ドロップ PHB
- **af23** : Assured Forwarding Class 2 High ドロップ PHB
- **af31** : Assured Forwarding Class 3 Low ドロップ PHB
- **af32** : Assured Forwarding Class 3 Medium ドロップ PHB
- **af33** : Assured Forwarding Class 3 High ドロップ PHB
- **af41** : Assured Forwarding Class 4 Low ドロップ PHB
- **af42** : Assured Forwarding Class 4 Medium ドロップ PHB
- **af43** : Assured Forwarding Class 4 High ドロップ PHB
- **be** : ホップ単位のベストエフォート型動作 (デフォルト)
- **cs1** : クラスセクタ 1 PHB
- **cs2** : クラスセクタ 2 PHB
- **cs3** : クラスセクタ 3 PHB
- **cs4** : クラスセクタ 4 PHB
- **cs5** : クラスセクタ 5 PHB
- **cs6** : クラスセクタ 6 PHB
- **cs7** : クラスセクタ 7 PHB
- **ef** : 低遅延トラフィック用の Expedited Forwarding (EF; 完全優先転送)

## スイッチオーバー移行の最適化

複数の SRP 設定オプションがあります。これにより、VoLTE トラフィックのサポートにおいて、アクティブゲートウェイからスタンバイゲートウェイ (主に P-GW) への移行時間が短縮されます。



**重要** これらの機能には、機能拡張をサポートするために、更新された ICSR ライセンスが必要です。詳細については、シスコのアカウント担当者にお問い合わせください。

### ICSR スwitchオーバー時の非 VoLTE トラフィックの許可

ICSR フレームワークは、課金情報を調整する前に、新しくアクティブになったゲートウェイ上で VoLTE トラフィックを有効にし、アカウントリングが重要ではないと判断された場合に新しくアクティブになったゲートウェイとの通信を有効にすることで、VoLTE トラフィックのスイッチオーバーの中断を軽減します。

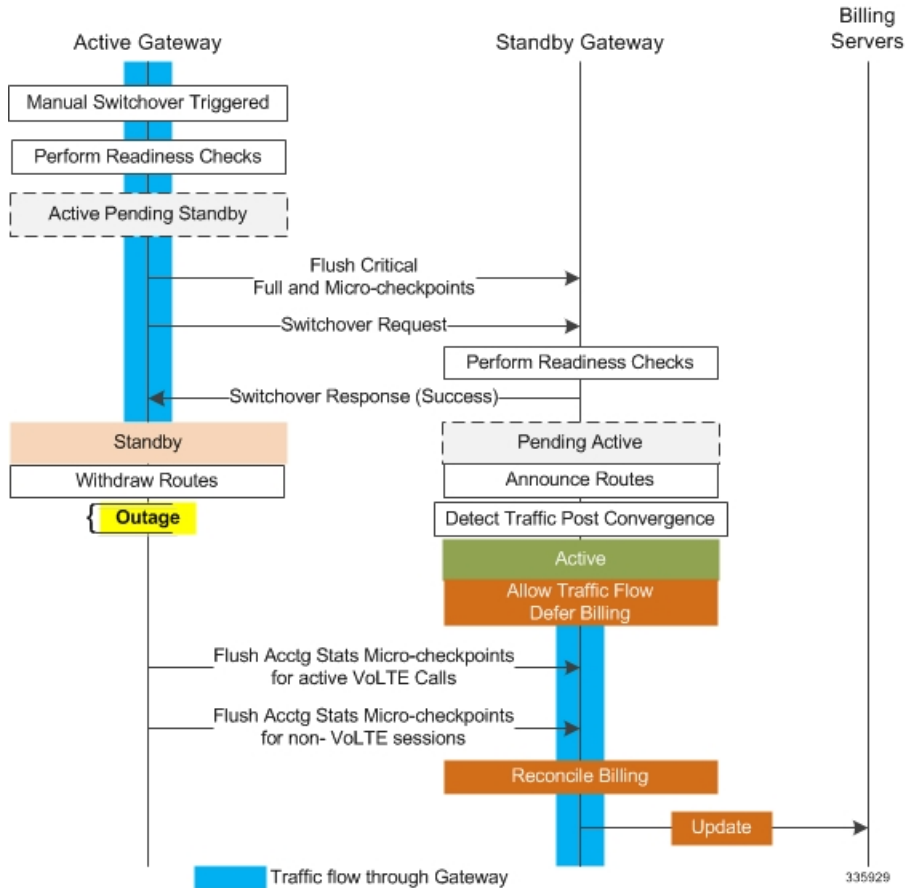
この機能は、データセッションや IMS/e911 のデフォルトのベアラートラフィックなど、他のすべてのトラフィックに拡張され、すべての非 VoLTE データトラフィックに対して次の ICSR 機能が提供されます。

- スwitchオーバーが発生すると、新しくアクティブになったゲートウェイは、ゲートウェイがアクティブになった時点ですべてのトラフィックを転送します。
- 課金サーバーとの外部通信は保留されています。次のトラフィックフロー図参照してください。



- 新しくアクティブになったゲートウェイは、以前のアクティブなゲートウェイから課金関連のチェックポイント情報をすべて受信すると、外部課金サーバー OCS（オンライン課金システム）または OFCS（オフライン課金システム）と通信する前に課金データを調整します。

図 11: コールフロー : 非 VoLTE データ停止の削減



SRP コンフィギュレーションモードの **switchover allow-all-data-traffic** CLI コマンドを使用すると、スイッチオーバーの移行中にすべてのデータトラフィック（VoLTE および非 VoLTE）が許可されます。このコマンドは、P-GW で有効になっている場合、**switchover allow-volte-data-traffic** コマンドを上書きします。

```
configure
context context_name
service-redundancy-protocol
switchover allow-all-data-traffic
```



**重要** この機能を有効にするには、**switchover allow-all-data-traffic** コマンドを両方のシャーシで実行する必要があります。

SRP コンフィギュレーションモードの **switchover allow-volte-data-traffic** CLI コマンドは、ICSR スイッチオーバーの移行中に VoLTE データトラフィックを許可します。

**configure**

```
context context_name
  service-redundancy-protocol
    switchover allow-volte-data-traffic [ maintain-accounting ]
```

注：

- **maintain-accounting** が有効になっている場合、VoLTE コールのアカウント精度が維持されます。VoLTE アカウンティングの統計情報がフラッシュされた後、アクティブなゲートウェイで VoLTE データが許可されます。

すべてのデータトラフィックの許可

SRP コンフィギュレーションモードの **switchover allow-all-data-traffic** コマンドを使用すると、スイッチオーバーの移行中にすべてのデータトラフィック (VoLTE および非 VoLTE) が許可されます。このコマンドは、P-GW で有効になっている場合、**switchover allow-volte-data-traffic** コマンドを上書きします。この機能により、スイッチオーバー中のデータトラフィックの停止を短くします。




---

**重要** この機能を有効にするには、この CLI コマンドがアクティブシャーシとスタンバイシャーシの両方で実行されている必要があります。

---

フラッシュおよび内部監査中は、すべてのデータトラフィックがアクティブシャーシで許可されます。フラッシュが完了すると、課金情報はバックグラウンドで調整されます。

早期にアクティブへの移行を許可

SRP コンフィギュレーションモードのスイッチオーバーの **allow-early-active-transition** コマンドを実行すると、ICSR スイッチオーバー中にアクティブ状態への早期の移行が可能になります。デフォルトでは、この機能はディセーブルになっています。

このコマンドは、計画されたスイッチオーバー時のデータ停止時間をさらに短縮するために、**allow-all-data-traffic** または **allow-volte-data-traffic** (アカウント精度の維持オプションは指定しない) コマンドと組み合わせて使用します。停止時間とは、ICSR スイッチオーバーを開始してから、新しくアクティブになったシャーシがデータの処理を開始するまでの時間です。




---

**重要** このコマンドを有効にする前に、両方の ICSR シャーシで上記のコマンドのいずれかを有効にする必要があります。

---

失敗したコールの監査後における ICSR の正常なクリーンアップ

セッションリカバリまたは ICSR イベント後のゲートウェイ (P-GW/S-GW/GGSN/SAE-GW) の監査中に、内部または外部でサブスクリバセッションに関連する重要な情報が矛盾していると思われる場合、ICSR は関連するセッション情報をローカルで消去します。

外部ゲートウェイ（ピアノード）はこのセッションの消去を認識していないため、UEセッションは他のノードで維持される場合があります。これにより、ゲートウェイの外部のリソースが占有され、VoLTE コール用の UE に到達できなくなります。

この機能を有効にすると、失敗したコールの ICSR 監査のために正常なクリーンアップが行われます。外部シグナリングは、セッションを消去する前にセッションの終了をピアに通知します。ゲートウェイは、セッションの削除を外部ピアに通知しようとしています。ローカルゲートウェイへの外部ノードには、S-GW、P-GW、SGSN、MME、AAA、PCRF および IMSA が含まれます。

監査エラーは、セッション情報が欠落しているか不完全であることが原因で発生する可能性があります。したがって、情報が使用可能なピアのみに通知されます。

グローバルコンフィギュレーションモードの **require graceful-cleanup-during-audit-failure** CLI コマンドは、正常なクリーンアップの機能を有効または無効にします。

```
configure
  require graceful-cleanup-during-audit-failure [ del-cause non-ims-apn
    { system-failure | none } ]
```

## スイッチオーバー制御の停止時間の最適化

ICSR フレームワークは、完全なチェックポイント統計情報のフラッシュ、ネットワークコンバージェンス、および内部監査に関連する制御停止時間を最小限に抑えます。

次のアクティビティによって消費される時間は、スイッチオーバー時の制御停止時間に影響します。

- **クリティカルフラッシュ**：アクティブから保留中への移行中に、すべての `sessmgrs` が保留中の重要な FC をフラッシュします（完全なチェックポイント）。この間、アクティブシャージはすべての制御パケットをドロップします。この段階で制御シグナリングが許可されている場合は、制御メッセージタイプに基づいてコールが切断され、アカウントリング情報が失われます。
- **ネットワークコンバージェンス**：ルートを更新し、新たにアクティブになったシャージに制御およびデータを送信するのにかかる時間が含まれます。制御メッセージは、移行中にドロップされます。
- **アカウントリングフラッシュ**：このフラッシュの段階で、シャージ間でデータカウントが同期されます。フラッシュ中に制御シグナリングが許可されている場合、コールは制御メッセージタイプに基づいて切断される可能性があり、スイッチオーバー前に存在したコールについてはアカウントリング情報が失われます。
- **監査**：コールリソースの同期によってコールがクリアされる可能性があるため、監査中に新しいコールは許可されません。

**switchover control-outage-optimization** CLI コマンドは、監査が完了するとすぐに、アカウントリングのフラッシュ中に新しいコールを許可します。この **SRP** コンフィギュレーションモードコマンドは、ICSR スwitchオーバー後の制御トラフィック（コールセットアップ、変更、削除）の迅速な復元を可能にします。

```
configure
  context context_name
    service-redundancy-protocol
```

```
switchover control-outage-optimization
end
```

## SRP コンテキストのインターフェイスパラメータの設定

この手順では、SRP コンテキスト内の IP アドレスとポート番号を使用して通信インターフェイスを設定します。このインターフェイスはシャーシ間通信をサポートします。



**重要** CLI コマンドは、両方のシャーシで実行する必要があります。続行する前に、両方のシャーシにログインします。常に、プライマリシャーシで設定の変更を最初に行います。

### configure

```
context vpn_ctxt_name [-noconfirm]
  interface srp_if_name
  ip-address { ip_address | ip_address/mask }
  exit
exit
port ethernet slot_num/port_num
  description des_string
  medium { auto | speed { 10 | 100 | 1000 } duplex { full | half } }
  no shutdown
  bind interface srp_if_name srp_ctxt_name
end
```

## SRP チェックポイント メッセージング エラーに対する NACK 生成の設定

### スタンバイシャーシからの NACK メッセージングの有効化

ICSR シャーシ間の SRP リンクでは、トランスポート (TCP) レベルの再送信がサポートされています。SRP の設定では、スタンバイシャーシでチェックポイントを実際に受信できるようにするために、オプションのアプリケーションレベルのチェックもサポートされています。チェックポイントの受信と適用が失敗すると、NACK メッセージがアクティブシャーシに送信されます。

この機能が有効になっていて、スタンバイシャーシが正常に適用されなかったマイクロチェックポイント (MC) の受信に回答して NACK を送信した場合、スタンバイシャーシは別の NACK を送信します。マクロチェックポイント (FC) が受信されない場合、スタンバイシャーシは 10 分間により多くの NACK (設定可能、デフォルトは 3) を送信します。FC が受信されて適用されるか、または設定された最大応答数に達するまで、NACK は送信され続け、10 分のリセットが行われます。

障害が発生した MC または FC への応答として、10 分間に NACK を送信する回数を指定することもできます (デフォルトは 3)。

SRP コンフィギュレーションモードの **checkpoint session** コマンドで **nack** キーワードを使用すると、スタンバイ ICSR シャーシのチェックポイントメッセージの障害に応じて、NACK メッセージの生成を有効にすることができます。



**重要** **nack** キーワードは、特別な ICSR 最適化機能のライセンスが購入およびインストールされている場合にのみ表示されます。詳細については、シスコのアカウント担当者にお問い合わせください。

```
configure
context context_name
  service-redundancy-protocol variable
    checkpoint session nack { macro | micro } [ max-response number
]
  no checkpoint session nack { macro | micro }
end
```

注：

- **max-response** は、障害が発生した MC または FC への応答として、10 分間に NACK が送信された回数で、0 ~ 65535 の整数で表されます（デフォルトは 3）。

SRP コンフィギュレーションモードの **checkpoint session** コマンドの **periodic-interval** キーワードを使用すると、アクティブシャーシとスタンバイシャーシ間のマクロチェックポイント (FC) の送信間隔を設定できます。



**重要** **eriodic-interval** キーワードは、特別な ICSR 最適化機能のライセンスが購入およびインストールされている場合にのみ表示されます。詳細については、シスコのアカウント担当者にお問い合わせください。

```
configure
context context_name
  service-redundancy-protocol variable
    checkpoint session periodic-interval minutes
    default checkpoint session periodic-interval
    no checkpoint session periodic-interval
end
```

## NACK メッセージングの選択的無効化

NACK メカニズムは、スタンバイシャーシで ICSR チェックポイント障害が発生した場合に、NACK メッセージを送信します。スタンバイシャーシから送信されたすべての NACK がアクティブシャーシからの完全なチェックポイントをトリガーします。

マイクロチェックポイントが継続的に失敗し、NACK を送信している場合、アクティブシャーシは完全なチェックポイントを送信し続けます。これにより、SRP の帯域幅が増加します。

CLI コマンドを使用すると、オペレータは特定のマイクロチェックポイントに対する NACK メッセージを選択的に無効にしたり、再度有効にしたりできます。

Exec モード **srp disable nack micro-chkpt-cmd** は、スタンバイシャーシからの NACK の送信を無効にします。

```
srp disable nack micro-chkpt-cmd chkpt_number
```

*chkpt\_number* は、無効にするチェックポイント番号を 1～255 の整数として指定します。 **show srp checkpoint info** コマンドの出力からチェックポイント番号 (CMD ID) を取得できます。

**srp enable nack micro-chkpt-cmd** コマンドを使用して、マイクロチェックポイントを再度有効にすることができます。

```
srp enable nack micro-chkpt-cmd chkpt_number
```

## LZ4 圧縮アルゴリズムの設定

必要に応じて、SRP メッセージングペイロードの LZ4 圧縮アルゴリズムを有効にすることができます。zlib アルゴリズムはデフォルトのままになります。

LZ4 は非常に高速な無損失圧縮アルゴリズムであり、マルチスレッドアプリケーションに対して直線的なスケーラビリティを備えています。

SRP コンフィギュレーションモードの **checkpoint session** コマンドの **compression** キーワードを使用すると、LZ4 圧縮アルゴリズムの使用を有効にすることができます。



**重要** **compression** キーワードは、特別な ICSR 最適化機能のライセンスが購入およびインストールされている場合にのみ表示されます。詳細については、シスコのアカウント担当者にお問い合わせください。

次のコマンドシーケンスは、LZ4 圧縮の使用を有効にします。

```
configure
  context context_name
    service-redundancy-protocol
      checkpoint session compression lz4
    end
```

LZ4 圧縮は、両方のシャーシが LZ4 を使用して設定されている場合にのみ有効です。いずれかのシャーシに zlib (デフォルト) が設定されている場合、圧縮アルゴリズムは zlib に戻ります。このアルゴリズムは、最初のソケットの確立時にのみネゴシエートされます。同意すると、TCP ソケット接続がリセットされるまでネゴシエーションは行われません。

## スタンバイの ICSR シャーシによる同期時間の短縮

SRP データベースを同期するためのデフォルトの方法では、アクティブなセッションマネージャとスタンバイセッションマネージャの間に TCP 接続が確立されるたびに、10 秒間の遅延が必要です。TCP 接続が確立されると、ハートビートメッセージが両方の ICSR シャーシ間で 3 秒ごとに交換されます。スタンバイシャーシは、データを受け入れる準備が整うまで、アクティブなシャーシから 7 つのハートビートメッセージを待機します。これにより、スタンバイシャーシでのセッションマネージャデータベースの同期が大幅に遅延する可能性があります。

セッションマネージャデータベースを同期するためのアグレッシブな方式を有効にすると、次のシナリオでのリカバリ時間が短縮されます。

- スタンバイセッションマネージャのクラッシュ
- スタンバイシャーシでのパケット処理カードの障害
- スタンバイシャーシの再起動
- SRP 接続の一時的な損失とリカバリ

アグレッシブな方式は、ICSR シャーシ間で交換されるハートビートメッセージの数とハウスキーピング情報の量を削減します。

SRP コンフィギュレーションモードの **standby database-recovery aggressive** コマンドを使用すると、SRP データベースの通常またはアグレッシブな復元を選択できます。

次のコマンドシーケンスは、アグレッシブリカバリモードを有効にします。

```
configure
  context context_name
    service-redundancy-protocol
      standby database-recovery aggressive
    end
```

このコマンドのデフォルト形式は、SRP データベースリカバリの通常モードを復元します。

## SRP 設定の確認

各シャーシで **show SRP info** コマンド (Exec モード) を実行して、SRP コンテキストが作成され、適切に設定されていることを確認します。

注：

- 間隔は、30～1440 の範囲で 15 で割り切れる整数で指定されます (デフォルトは 45 分)。完全なチェックポイントを送信する間隔の範囲は、30 分～24 時間 (1440 分) です。

## ICSR の送信元コンテキストの変更

コアサービスの送信元コンテキストを変更するには、次のようにします。

- 
- ステップ 1** ボーダー ゲートウェイ プロトコル (BGP) のルータの AS パスを追加し、[BGP ルータとゲートウェイアドレスの設定 \(400 ページ\)](#) の設定例を適用して、コアネットワークサービスが設定されている送信元コンテキストのゲートウェイ IP アドレス、ネイバー IP アドレス、リモート IP アドレスを設定します。
  - ステップ 2** [BGP 用の SRP コンテキストの設定 \(400 ページ\)](#) の設定例を適用して、BGP ネイバーコンテキストと IP アドレスを使用してサービス冗長性コンテキストを設定し、BGP リンクアクティビティをモニターします。
  - ステップ 3** [BGP 設定の確認 \(401 ページ\)](#) の手順に従って、BGP コンテキストの設定を確認します。
  - ステップ 4** 「設定の確認と保存」の説明に従って、設定を保存します。
-

## BGP ルータとゲートウェイアドレスの設定

BGP コンテキストとネットワークアドレスを作成するには、次の例を使用します。

```
configure
context source_ctxt_name
router bgp AS_num
network gw_ip_address
neighbor neighbor_ip_address remote-as AS_num
end
```

注：

- *source\_ctxt\_name* は、コアネットワークサービスが設定されているコンテキストです。

## BGP 用の SRP コンテキストの設定

次の例を使用して、SRP コンテキストでの BGP コンテキストと IP アドレスを設定します。

```
configure
context srp_ctxt_name
service-redundancy-protocol
monitor bgp context source_ctxt_name neighbor_ip_address
end
```

*neighbor\_ip\_address* は、IPv4 のドット付き 10 進表記または IPv6 のコロンで区切られた 16 進表記で入力できます。コンテキストごとに、IPv4 アドレスまたは IPv6 IP アドレスとして複数の IP アドレスを追加できます。

ICSR フェールオーバーは、コンテキスト内のすべての BGP ピアがダウンしたときにトリガーされます。

必要に応じて、コンテキスト内に SRP ピアグループを設定できます。ICSR フェールオーバーは、グループ内のすべてのピアで障害が発生した場合に発生します。このオプションは、IPv4 と IPv6 のピアの組み合わせが複数ペアの VLAN に分散され、ピアグループのすべてのメンバーによって IPv4 または IPv6 の接続が失われる展開で役立ちます。

コンテキスト内の SRP ピアグループの設定例（「PGWin」）を次に示します。

```
monitor bgp context PGWin 10.1.1.16 group 1
monitor bgp context PGWin 10.1.1.17 group 1
monitor bgp context PGWin 69.2.215.0 group 2
monitor bgp context PGWin 69.2.215.1 group 2
monitor bgp context PGWin 2001:4333:201:1102:103:2a1:: group 3
monitor bgp context PGWin 2001:4333:201:1102:103:2a1:0:1 group 3
```

上記の設定例では、グループ 1、2、または 3 のすべてのアドレスで接続を失った場合、ICSR フェールオーバーが発生します。

詳細については、『*Command Line Interface Reference*』と「*Service Redundancy Protocol Configuration Mode Commands*」の章の **monitor bgp**、**monitor diameter**、および **monitor authentication-probe** の各コマンドの説明を参照してください。



## BGP 設定の確認

`show srp monitor bgp` コマンド (Exec モード) を入力して、BGP 設定を確認します。

## ICSR の接続先コンテキストの変更

コアサービスの接続先コンテキストを変更するには、次の手順を実行します。

- ステップ 1** BGP ルータを追加し、[接続先コンテキストでの BGP ルータとゲートウェイアドレスの設定 \(401 ページ\)](#) の設定例を適用してコアネットワークサービスが設定されている接続先コンテキストのゲートウェイ IP アドレス、ネイバー IP アドレス、リモート IP アドレスを設定します。
- ステップ 2** BGP ネイバーコンテキストと IP アドレスを使用してサービス冗長性コンテキストを設定し、[接続先コンテキスト用の BGP の SRP コンテキストの設定 \(401 ページ\)](#) の設定例を適用して BGP リンクアクティビティをモニターします。
- ステップ 3** [サブスクリバをデフォルトモードに設定 \(402 ページ\)](#) の手順に従って、サブスクリバモードを **default** に設定します。
- ステップ 4** [接続先コンテキストでの BGP 設定の確認 \(402 ページ\)](#) の手順に従って、BGP コンテキストの設定を確認します。
- ステップ 5** 「設定の確認と保存」の説明に従って、設定を保存します。

## 接続先コンテキストでの BGP ルータとゲートウェイアドレスの設定

BGP コンテキストとネットワークアドレスを作成するには、次の例を使用します。

```
configure
context dest_ctxt_name
router bgp AS_num
network gw_ip_address
neighbor neighbor_ip_address remote-as AS_num
end
```

注：

- `AS_num` は、この BGP ルータの自律システムのパス番号です。

## 接続先コンテキスト用の BGP の SRP コンテキストの設定

次の例を使用して、SRP コンテキストでの BGP コンテキストと IP アドレスを設定します。

```
configure
context srp_ctxt_name
service-redundancy-protocol
monitor bgp context dest_ctxt_name neighbor_ip_address
end
```

## サブスクリイバをデフォルトモードに設定

サブスクリイバのモードをデフォルトに設定するには、次の例を使用します。

```
configure
  context dest_ctxt_name
    subscriber default
  end
```

## 接続先コンテキストでの BGP 設定の確認

`show srp monitor bgp` コマンド (Exec モード) を入力して、BGP 設定を確認します。

## スタンバイシステムでのバルク統計情報収集の無効化

システムからのバルク統計情報の収集は、スタンバイ動作モードのときに無効にすることができます。



**重要** この機能が有効になっていて、システムがスタンバイ状態に移行すると、保留中の累積統計データが最初の機会に転送されます。その後、システムがスタンバイ状態から抜け出すまで、追加の統計情報の収集は行われません。

スタンバイシステムでバルク統計情報収集を無効にするには、次の例を使用します。

```
configure
  bulkstat mode
    no gather-on-standby
  end
```

この手順を両方のシステムで繰り返します。

## プライマリ設定とバックアップ設定の確認

この項では、プライマリシステムとバックアップシステムの ICSR 設定の比較方法について説明します。

**ステップ 1** 各システム (Exec モード) で `show configuration srp` コマンドを入力します。

**ステップ 2** 両方のシャーシの SRP 設定情報が同じであることを確認します。

出力は次のようになります。

```
config
  context source
    interface haservice loopback
      ip address 172.17.1.1 255.255.255.255 srp-activate
    #exit
    radius attribute nas-ip-address address 172.17.1.1
    radius server 192.168.83.2 encrypted key 01abd002c82b4a2c port 1812
    radius accounting server 192.168.83.2 encrypted key 01abd002c82b4a2c port 1813
```

```
ha-service ha-pdsn
  mn-ha-spi spi-number 256 encrypted secret 6c93f7960b726b6f6c93f7960b726b6f hash-algorithm
md5
  fa-ha-spi remote-address 192.168.82.0/24 spi-number 256 encrypted secret 1088bdd6817f64df
  bind address 172.17.1.1
#exit
#exit
context destination
  ip pool dynamic 172.18.0.0 255.255.0.0 public 0 srp-activate
  ip pool static 172.19.0.0 255.255.240.0 static srp-activate
#exit
context srp
  service-redundancy-protocol
#exit
#exit
```

## サブスクリバ状態管理の監査プロセスの設定

この監査は、2つの ICSR ピアが同期していることを確認し、スケジュール済みまたは予定外のスイッチオーバーイベントの前に不一致を識別します。

- ステップ 1** SRP コンテキストモードを開始し、**service-redundancy-protocol** コマンドを入力します。
- ステップ 2** **audit daily-start-time** コマンドを入力します。毎日の開始時刻を時間と分で指定します。たとえば、06 00 の開始時刻は、監査が 6:00 AM に開始されることを示しています。
- ステップ 3** **audit periodicity** コマンドを入力します。SRP 監査統計情報を生成する間隔を、60 ~ 1440 の整数として分単位で指定します。たとえば、90 の周期は、指定された開始時間以降に、SRP 監査統計情報が 90 分ごとに生成されることを示します。デフォルト = 60 です。  
設定シーケンスの例を次に示します。

```
config
  context srp
    service-redundancy-protocol
      audit daily-start-time 06 00
      audit periodicity 90
    end
```

## ICSR 運用のトラブルシューティング

### SSD

StarOS は、Exec モードの一連の **show** コマンドの結果を出力する ICSR 固有の **show support details** (SSD) コマンドをサポートしています。このミニ SSD によって、アクティブとスタンバイのシャーシ間で ICSR のタイミングに関する問題をデバッグする際のキャプチャ時間を短縮し、問題の迅速な解決を助長します。

**show support details icsr** コマンドは、次の **show** コマンドの出力を含むミニ SSD を生成します。

- show srp info
- show srp checkpoint statistics
- show srp checkpoint statistics verbose
- show srp checkpoint statistics debug-info
- show srp checkpoint statistics sessmgr all
- show srp checkpoint statistics sessmgr all debug-info
- show srp checkpoint statistics ipsecmgr all
- show srp checkpoint statistics sessmgr all write-list-stats
- show srp checkpoint info
- show srp monitor
- show srp monitor all
- show srp monitor diameter debug
- show srp statistics
- show srp call-loss statistics
- show srp audit-statistics
- show session subsystem facility sessmgr all debug-info

SSD の出力は、**/flash** またはシャージ外に保存できるファイルに転送できます。詳細については、『*Command Line Interface Reference*』を参照してください。

### show srp details

Exec モードの **show srp details** コマンドは、ICSR/SRP の問題をトラブルシューティングするために TAC 担当者が使用する包括的な情報を表示します。

## オペレーティングシステムの更新

ICSR システム上のオペレーティングシステム (StarOS™) の更新は、スタンバイモードの間にシステムごとに個別に実行されます。スタンバイシステムが更新されている間、アクティブなシステムがコールセッションを処理するため、トラフィックの中断は最小限に抑えられます。

一般的な更新手順は次のとおりです。

1. StarOS のソフトウェアイメージをダウンロードし、アクティブとスタンバイの両方のシステムにコピーまたは転送します。
2. 両方のシステムで現在実行中の設定を保存します。
3. スタンバイ バックアップ システムを最初に更新します。
4. スタンバイ バックアップ システムがアクティブになるように、アクティブなプライマリシステムから SRP スイッチオーバーを開始します。
5. スタンバイ プライマリ システムを更新します。

6. スタンバイプライマリシステムがアクティブになるように、アクティブなバックアップシステムから SRP スイッチオーバーを開始します。

次に、StarOS のアップグレードプロセスを完了するために必要なすべての手順を詳しく説明した、4 つのパートから成るフローチャートを示します。

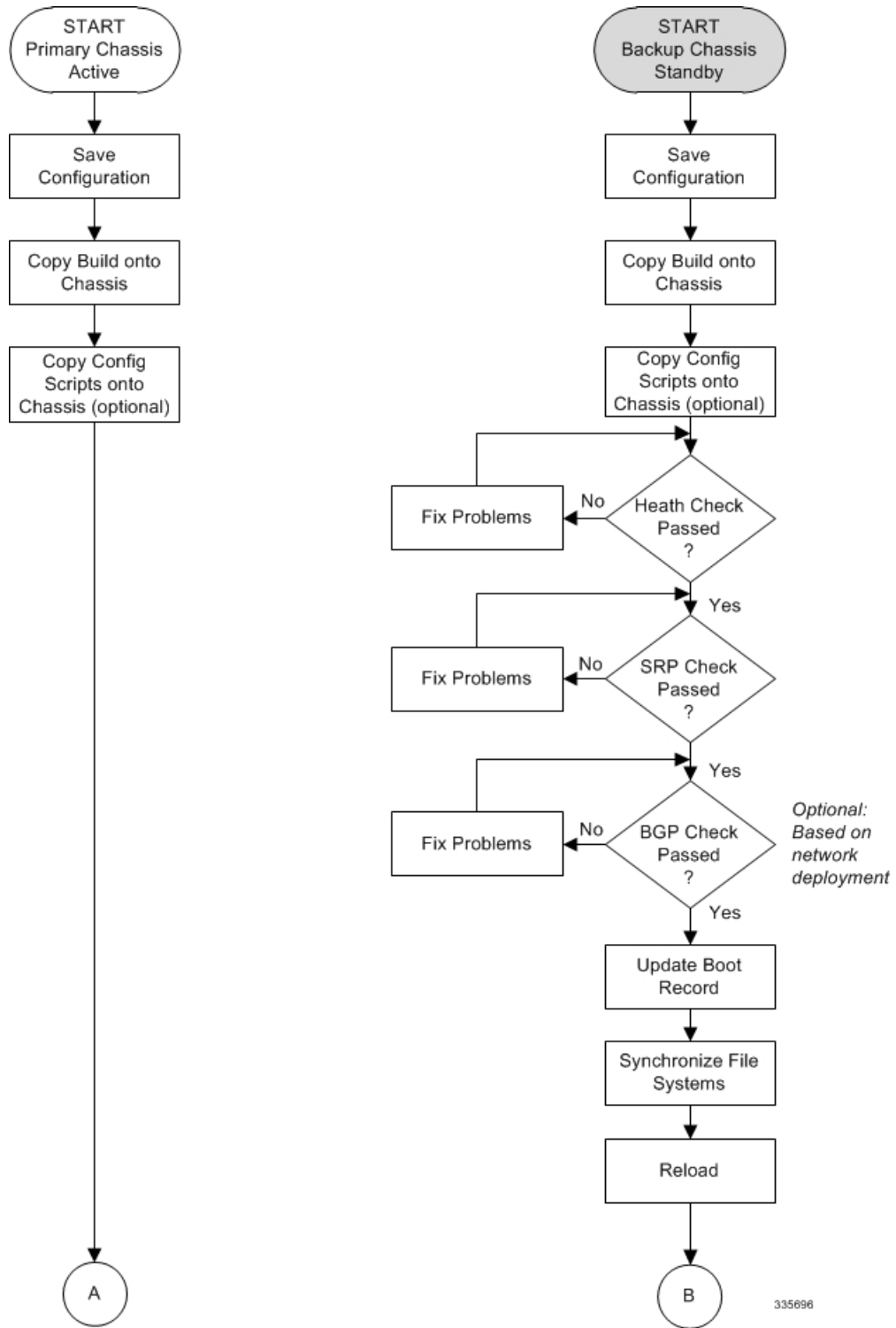


---

**注意** MIO/UMIO/MIO2 機能でデマルチプレクサを有効にすると、システム内のリソース割り当てが変更されます。これは、ICSR 設定内の StarOS バージョン間のアップグレードまたはダウングレードに直接影響を与えます。Star OS 21.24 リリースでは、2 つのリリース (N-2) より上の ICSR アップグレードは完全には認定されていません。詳細については、ICSR の展開をアップグレードまたはダウングレードする前に、シスコのアカウント担当者に連絡して手順についてのサポートを受けてください。

---

図 12: ICSR ソフトウェアアップグレード : パート 1



335696

図 13: ICSR ソフトウェアアップグレード : パート2

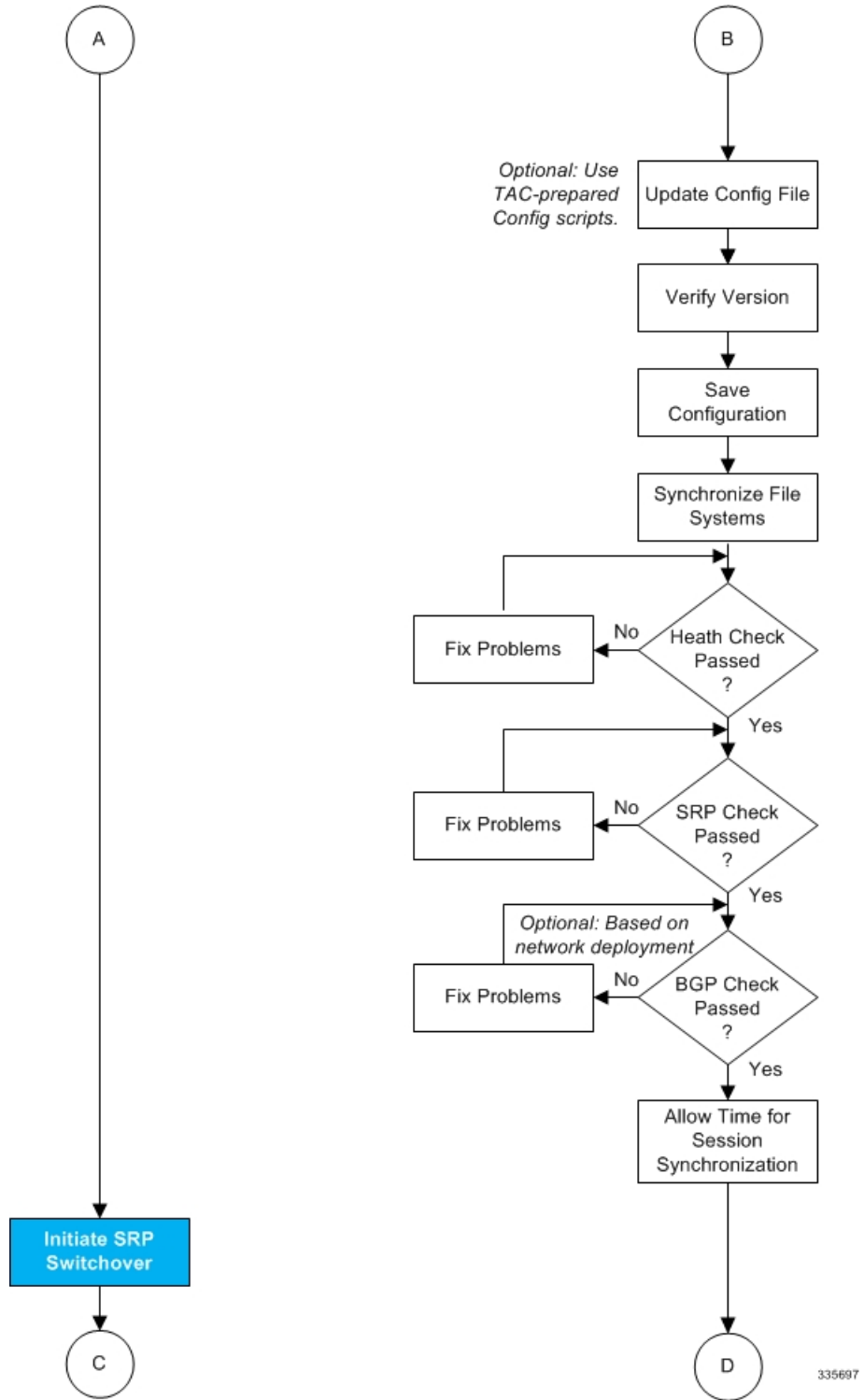
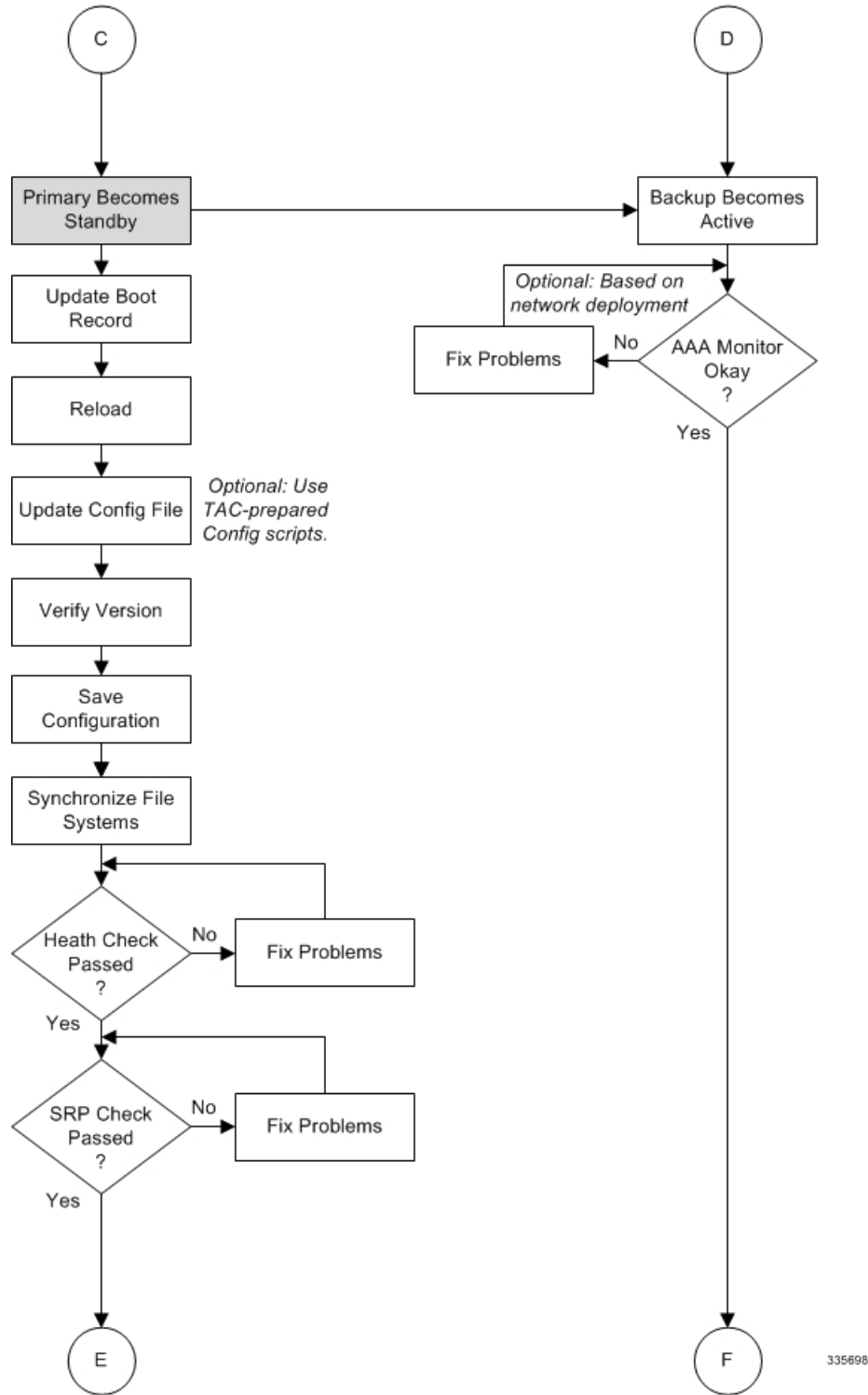


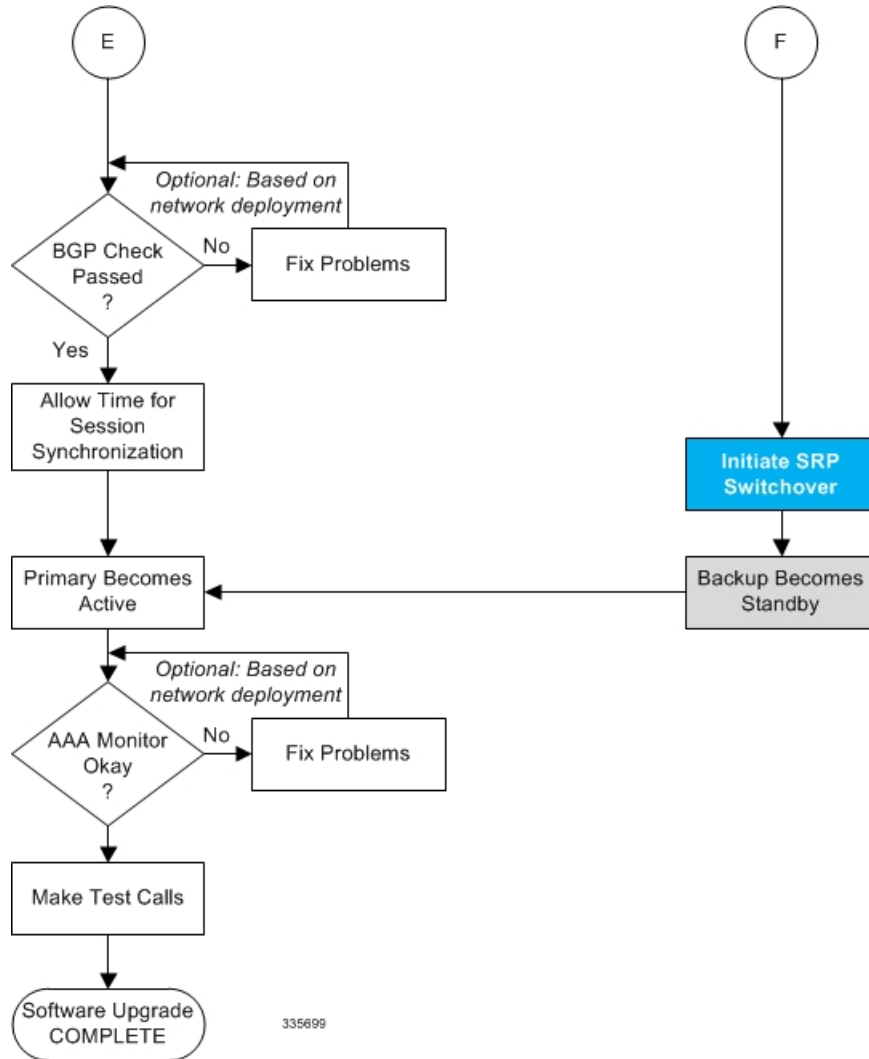
図 14: ICSR ソフトウェアアップグレード : パート3



335698



図 15: ICSR ソフトウェアアップグレード : パート 4



## 両方の ICSR システム

プライマリ（アクティブ）とバックアップ（スタンバイ）の両方の ICSR システムで、次に説明するタスクを実行します。

### StarOS イメージのダウンロードと転送

**ステップ 1** 次の Exec モードのコマンドを入力して、新しいオペレーティングシステムのイメージファイルに対応するために、**/flash** デバイスに十分な空き領域があることを確認します。

```
[local]host_name directory /flash
```

**ステップ 2** シスコのサポートサイトとダウンロードファシリティへのアクセスは、ユーザー名とパスワードで制御されています。**/flash** デバイスにアップロードできるネットワークの場所またはローカルドライブにソフトウェアイメージをダウンロードします。

**ステップ 3** 次のいずれかの方法を使用して、新しいオペレーティングシステムのイメージファイルを **/flash** デバイスに転送します。

a) `copy` コマンドを使用して、ネットワークの場所またはローカルドライブからファイルをコピーします。

```
[local]host_name copy from_url to_url [-noconfirm]
```

b) システムへのアクセス権を持つ FTP クライアントを使用して、ファイルを **/flash** デバイスに転送します。FTP クライアントは、バイナリモードを使用してファイルを転送するように設定する必要があります。

c) システムへのアクセス権を持つ SFTP クライアントを使用して、ファイルを **/flash** デバイスに転送します。

**ステップ 4** 次の Exec モードのコマンドを実行して、イメージファイルが **/flash** デバイスに正常に転送されたことを確認します。

```
[local]host_name directory /flash
```

**ステップ 5** `show version /flash/image_filename` コマンドを実行して、ビルド情報を確認します。次に例を示します。

```
local]host_name show version /flash/image_filename.bin
```

(注) CRC エラーは、上記のコマンドの出力に表示されます。エラーが表示された場合は、ビルドを確認し、シャージに再転送します。正しいイメージバージョンとビルドの説明が表示されていることを確認します。

## スタンバイ ICSR システム

バックアップ ICSR またはスタンバイ ICSR システムで、次に記述するタスクを実行します。

### 正常性チェックの実行

正常性チェックは、ソフトウェアの更新を処理するためのシステムの準備状況を確認する Exec モードの一連の `show` コマンドです。

**ステップ 1** `show card table all |grep unknown` を実行します。出力は表示されません。

**ステップ 2** `show card table |grep offline` を実行します。出力は表示されません。

**ステップ 3** `show resources |grep Status` を実行します。出力に「Within acceptable limits」と表示されます。

**ステップ 4** `show alarm outstanding` を実行します。ソフトウェア更新の実行を妨げる可能性のある問題の有無を出力で確認します。

## SRP チェックの実施

サービス冗長性プロトコル (SRP) のチェックは、ICSR システムステータスのモニタリングメカニズムが動作可能であることを確認します。

**ステップ 1** `show srp monitor all` を実行します。

**ステップ 2** ソフトウェア更新の実行を妨げる可能性のある問題の有無を出力で確認します。

## BGP チェックの実施

ボーダー ゲートウェイ プロトコル (BGP) のチェックは、BGP を使用して冗長シャーシ間通信をサポートする場合にのみ必要です。これらのチェックは、コンテキストごとおよびサービスタイプごとに実行されます。

**ステップ 1** 各 BGP 対応コンテキストに対して、`show ip bgp summary` を実行します。BGP ピアが接続されており、IPv4 および IPv6 ピアが稼働していることを確認します。すべての BGP 対応コンテキストに対してこの手順を繰り返します。

**ステップ 2** `show service_name all |grep "Service Status:"` を実行します。サービスを「開始」する必要があります。シャーシで実行されているすべてのサービスに対して手順を繰り返します。

## ブートレコードの更新

最近ダウンロードしたソフトウェアイメージ (.bin) ファイルの新しい起動スタックエントリを追加する必要があります。

**ステップ 1** Exec モードの `show boot` コマンドを実行して、`boot.sys` ファイルに 10 未満のエントリがあることと、高い優先順位のエントリが使用可能である (最小で起動スタックに優先順位 1 のエントリがない) ことを確認します。

**ステップ 2** 次のグローバルコンフィギュレーションコマンドを入力して、新しいオペレーティングシステムイメージファイルと現在使用されている CLI 設定ファイルで構成される新しいファイルグループの新しい起動スタックエントリを作成します。

```
[local]host_name(config)# boot system priority number image image_url /flash/filename config
cfg_url /flash/filename
```

**ステップ 3** <N-1> 方式を使用して、このエントリに次に高い優先順位を割り当てます。この場合、優先順位番号は、現在の最高の優先順位よりも 1 つ小さい値を割り当てます。

優先順位 1 が使用されている場合は、既存のエントリの数字を再割り当てし、少なくともその優先順位 1 を使用できるようにします。

`boot.sys` ファイルに含めることができる起動スタックエントリの最大数は 10 です。起動スタックにすでに 10 個のエントリがある場合は、これらのエントリのうち少なくとも 1 つを削除する必要があります (通常

は優先順位が最も低いエントリ)。また、必要に応じて、他のエントリの一部またはすべての番号を再割り当てしてから続行します。起動スタックエントリを削除するには、**no boot system priority** コマンドを使用します。

**boot system priority** コマンドの使用方法については、このガイドの「新しい起動スタックエントリの追加」の項を参照してください。

## ファイルシステムの同期

次の Exec モードコマンドを入力して、ローカルファイルシステムを同期します。

```
[local]host_name# filesystem synchronize all
```

## StarOS の再起動

次のコマンドを入力して、StarOS を再起動します。

```
[local]host_name# reload [-noconfirm]
```

システムが再起動すると、以前に設定した新しい起動スタックエントリを使用して、新しいオペレーティングシステムのソフトウェアイメージとそれに対応する CLI コンフィギュレーションファイルがロードされます。

システムが再起動したら、CLI セッションを確立し、**show version** コマンドを入力して、アクティブなソフトウェアバージョンが正しいことを確認します。

**PDSN** のオプション：アップグレード中に IP プール共有プロトコルを使用している場合は、『*PDSN Administration Guide*』の「*Configuring IPSP Before the Software Upgrade*」を参照してください。

## 設定ファイルの更新

新しいオペレーティングシステムの機能によっては、コンフィギュレーションファイルの変更が必要になる場合があります。これらの変更は、手動で行うことも、Cisco TAC に用意されたカスタムスクリプトによって容易に行うこともできます。更新されたコンフィギュレーションファイルを保存する前に、必要な変更を行います。

## ソフトウェアバージョンの確認

システムが正常に起動したら、Exec モードの **show version** コマンドを実行して、新しい StarOS バージョンが実行されていることを確認します。

Exec モードの **show build** コマンドを実行すると、StarOS ビルドのリリースに関する追加情報を表示できます。

## 設定ファイルの保存

Exec モードの **save** 設定コマンドを使用して、現在の実行時設定を **/flash** デバイスとシャーシ外の場所（外部メモリデバイスやネットワーク URL）に保存します。シャーシ外コピーが、障害が発生した場合に使用できるフォールバック用のロード可能な設定ファイルとなります。

## 更新プロセスの完了

スタンバイセカンダリシャーシのアップグレードプロセスを完了するには、次のタスクを繰り返します。

- [ファイルシステムの同期 \(412 ページ\)](#)
- [正常性チェックの実行 \(410 ページ\)](#)
- [SRP チェックの実施 \(411 ページ\)](#)
- [BGP チェックの実施 \(411 ページ\)](#)

## セッションの同期の待機

次の手順に進む前に、ICSR シャーシ間で実行されるシステム同期の時間を割り当てます。

- 
- ステップ 1** 両方のシャーシで **how session recovery status verbose** コマンドを実行します。このコマンドの出力にエラーが表示されない場合にのみ、次の手順に進みます。
  - ステップ 2** スタンバイシャーシで **show srp checkpoint statistics |more** を実行します。
  - ステップ 3** アクティブシャーシで **show subs summary |grep Total** を実行します。
  - ステップ 4** アクティブシャーシのサブスライバの数と、スタンバイシャーシの現在の事前に割り当てられたコール数 (Current pre-allocated calls:) を比較します。これらは同様 (5% 以内) である必要があります。システムが同期を完了するまで数分間待ちます。
- 

## プライマリシステム (Primary System)

プライマリ (アクティブ) ICSR システムで次に説明するタスクを実行します。

### SRP スイッチオーバーの開始

SRP スイッチオーバーでは、プライマリシャーシがスタンバイモードになり、バックアップシャーシがアクティブになります。セカンダリシャーシが、アップグレードされたソフトウェアとのセッションを処理するようになりました。

- 
- ステップ 1** プライマリシャーシで、**srp initiate-switchover** コマンドを実行します。既存のすべてのセッションは、バックアップシャーシに移行され、新しいセッション要求の処理が開始されます。スイッチオーバープロセスが完了するのを許可します。
  - ステップ 2** プライマリシャーシで、**show srp info** コマンドを実行します。スイッチオーバーが完了すると、シャーシの状態は [Standby] を示します。
  - ステップ 3** バックアップシャーシで、**show srp info** コマンドを実行して、スイッチオーバーが完了していることを確認します。スイッチオーバーが完了すると、シャーシの状態は [Active] を示します。
-

## 新しくアクティブになったシステムでの AAA モニターステータスの確認

ネットワーク展開で AAA サーバーとの通信が必要な場合は、新しくアクティブになったシステムにログインし、AAA モニターチェックを実行します。システムが AAA サーバーと通信できないことを示す SNMP トラップ (**starSRPAAAUnreachable**) が存在するかどうかを確認します。

- 
- ステップ 1** Exec モードコマンドの **show snmp trap history |grep starSRPAAAUnreachable** を実行します。
- ステップ 2** このコマンドの出力がないか、または (イベントタイムスタンプに基づいて) 最新の SNMP トラップ通知がないはずです。
- ステップ 3** アクティブなシステムが 1 つ以上の AAA サーバーと通信できない場合、詳細については「[AAA モニター](#)」を参照してください。
- 

## ソフトウェア更新の実行

バックアップ (スタンバイ) システムにログインし、次のタスクを繰り返してバックアップ (スタンバイ) システムのアップグレードプロセスを実行します。

- ブートレコードの更新 (411 ページ)
- StarOS の再起動 (412 ページ)
- 設定ファイルの更新 (412 ページ)
- ソフトウェアバージョンの確認 (412 ページ)
- 設定ファイルの保存 (412 ページ)
- ファイルシステムの同期 (412 ページ)
- 正常性チェックの実行 (410 ページ)
- SRP チェックの実施 (411 ページ)
- BGP チェックの実施 (411 ページ)
- セッションの同期の待機 (413 ページ)

## SRP スイッチオーバーの開始

この SRP スイッチオーバーは、プライマリシステムをアクティブモードにして、バックアップシステムをスタンバイに戻します。プライマリシャーシが、アップグレードされたソフトウェアとのセッションを処理するようになりました。

- 
- ステップ 1** バックアップシャーシで、**srp initiate-switchover** コマンドを実行します。既存のすべてのセッションは、新しいセッション要求の処理を開始するプライマリシャーシに移行されます。スイッチオーバープロセスが完了するのを許可します。
- ステップ 2** バックアップシステムで、**show srp info** コマンドを実行します。スイッチオーバーが完了すると、シャーシの状態は [Standby] を示します。

**ステップ3** プライマリシステムで、**show srp info** コマンドを実行して、スイッチオーバーが完了していることを確認します。スイッチオーバーが完了すると、シャーシの状態は [Active] を示します。

---

## テストコールの実施

シャーシの状態が確認され、サブスライバが移行されたら、新しい発信テストを実行してコールが成功したことを確認します。

## フォールバック手順

以前の設定とソフトウェアビルドに戻すには、管理者権限を持つユーザーが次の手順を実行します。

---

**ステップ1** Exec モードの **show boot** コマンドを実行します。表示される最上位の最小番号のエントリは、新しいソフトウェアビルドが搭載された新しい設定である必要があります。2 番目の最上位エントリはバックアップ設定である必要があります。

**ステップ2** 最上位の起動エントリ **n** を削除し、管理カード間で設定を同期します。

```
[local]host_name# config
[local]host_name(config)# no boot system priority n
[local]host_name(config)# end
[local]host_name# filesystem synchronize all
```

**ステップ3** システムを再起動して、以前の設定をロードします。

```
[local]host_name# reload
```

**ステップ4** の説明に従って、正常性チェックを実行します。 [正常性チェックの実行 \(410 ページ\)](#)

---







## 第 28 章

# データコレクターのサポート

サポートデータコレクター (SDC) は、プロセス状態、カウンタ、イベント、および属性データのスケジュールされた収集を可能にするシステム機能であり、設置場所での問題のトラブルシューティングに役立つ場合があります。

この章は、次の項で構成されています。

- [概要 \(417 ページ\)](#)
- [SDR 収集の設定 \(418 ページ\)](#)
- [SDR 収集設定の表示 \(418 ページ\)](#)
- [SDR 情報の収集と保存 \(419 ページ\)](#)
- [レコード収集の管理 \(419 ページ\)](#)
- [SDR を使用した問題の診断 \(421 ページ\)](#)
- [SDR CLI コマンド \(421 ページ\)](#)

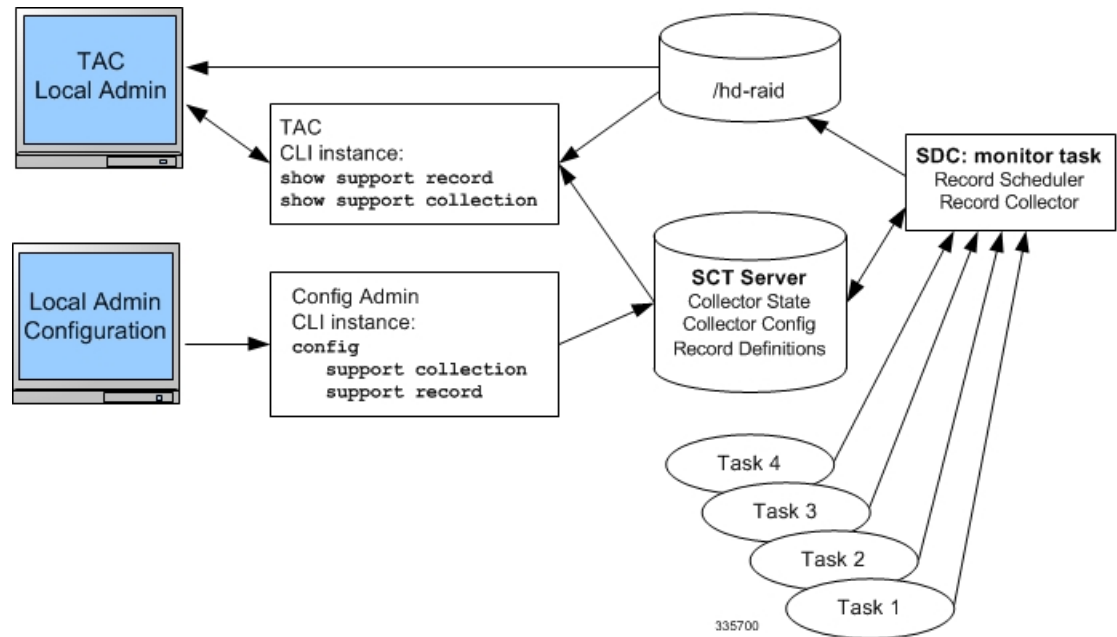
## 概要

サポートデータを収集するタスクは、`record collector` と呼ばれるバックグラウンド CLI タスクによって実行されます。管理者は、CLI を介して SDC を設定し、コマンドを定期的に行います。レコードコレクタは常にバックグラウンドで実行され、収集レコードがあるかどうかを確認します。

サポートデータを収集する時間になると、スケジューラは設定された CLI コマンドのシーケンスを実行し、その結果をハードディスク上の `gunzipped (gz)` ファイルに保存します。このファイルは SDR (サポートデータレコード) と呼ばれ、その時点でのシステム全体の状態のスナップショットを表します。

テクニカルアシスタンスセンター (TAC) 担当者およびローカル管理者は、SDR をオンラインで、またはシステムから転送して確認することができます。また、コレクタの状態の情報を調査する場合があります。次の図は、状態とカウンタの情報を含むシステムタスクを示しています。タスクとプロセスの間の矢印は、メッセージの要求を表し、データの主要なフローを示します。

図 16: SDC のタスクとプロセス &lt;



## SDR 収集の設定

サポートデータレコード (SDR) は、スタンドアロンの圧縮ファイルに保存され、順序が設定された CLI `support` コマンドの一連の表示出力です。各 CLI `support` コマンドの出力は、独自のレコードセクションに保存されます。レコードセクションは、レコードセクションの名前とその ASCII コマンドシンタックスで識別されます。たとえば、レコードセクション `show_version` では、CLI コマンド文字列「`show version`」が表示されます。

設定内にレコードセクションのコマンドが表示される順序は重要です。すべてのサポートレコードコマンドは、順序が設定されたセットとしてまとめて設定する必要があります。つまり、1つのコマンドのみを単独で指定するだけで、SDR 全体の内容を示す1つのコマンド出力になります。

ユーザーは、SDR の一連の特定のレコードセクションを設定できます。これには、デフォルトの SDR レコードセクションの一部またはすべてを含めることも、含めないこともできます。この設定は、設定ファイルの [Global Configuration] のセクションに保存されます。**support record section** コマンドの詳細については、[設定コマンド \(グローバルコンフィギュレーションモード\)](#) (422 ページ) を参照してください。

## SDR 収集設定の表示

ユーザーが `[support record]` セクションを指定していない場合は、**show configuration verbose** コマンドを実行すると、デフォルトの `[support record]` セクションが表示されます。ユーザーが `[support record]` セクションを設定している場合は、**show configuration** コマンドを実行すると、

ユーザーが設定した [support record] セクションが表示されます。サポート収集スケジュールの設定は、[Global Configuration] セクションの **show configuration** 出力にも表示されます。

## SDR 情報の収集と保存

スケジュールされた時刻に、サポートデータコレクター (SDC) がアクティブな場合はバックグラウンドで実行され、指定されたすべての **record section** コマンドが収集されます。この情報は、1つの連続した出力として連結されます。出力は圧縮され、**/hd-raid/support/record/** ディレクトリ内のディスクにファイルとして保存されます。

SDC の周期は、グローバル コンフィギュレーション モードで **support collection schedule** コマンドによって設定されます。SDR が保存されると、SDC は、**support collection** コマンドを介して指定されたスリープ時間間隔を待ってから、別の SDR を収集します。



---

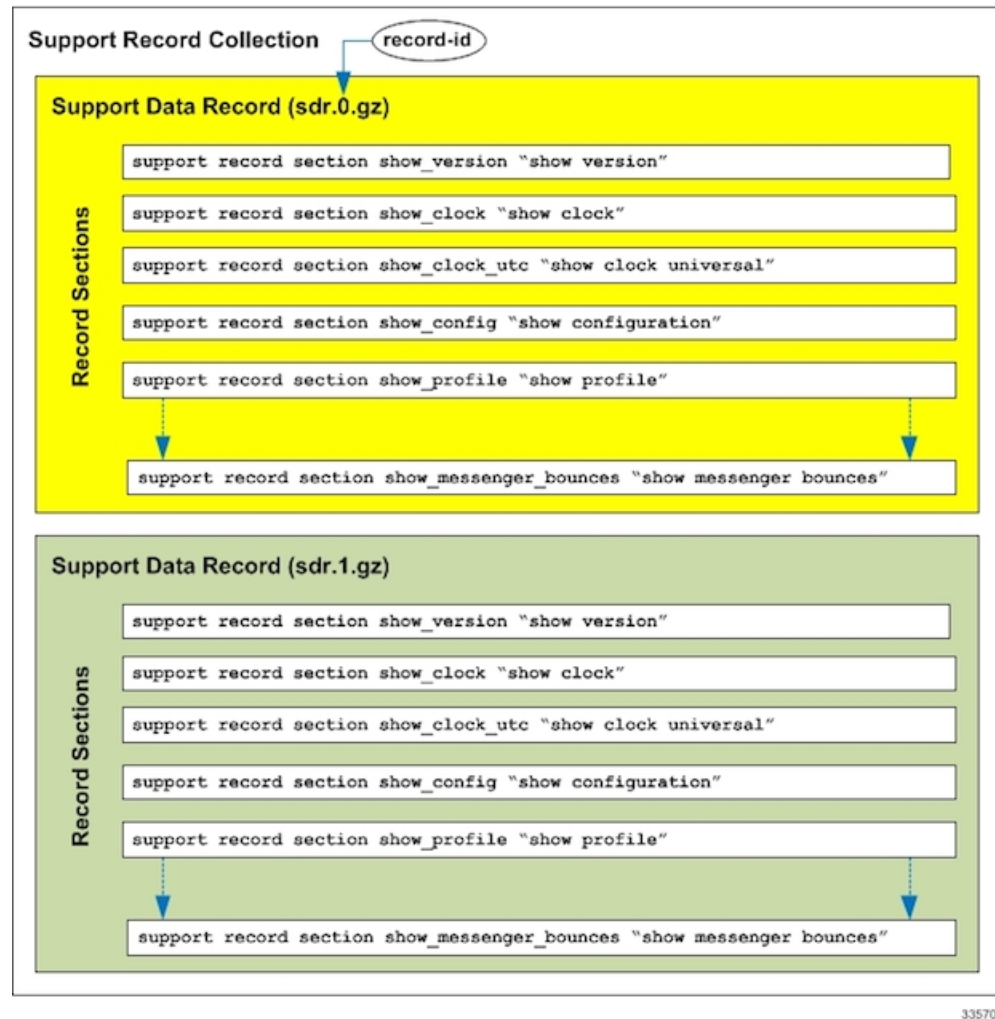
**重要** SDR 間の期間は、設定されているスリープ時間間隔と前のレコードの収集にかかった時間の合計と同じです。

---

## レコード収集の管理

SDR は、自己相対セットにまとめて保存されます。この自己相対セットは、サポートレコード収集と呼ばれます。個々の SDR は、**record-id** で識別されます。最新の SDR の **record-id** は常に 0 (ゼロ) です。次に古い SDR は **record-id 1** となり、保存されたコレクション内のレコード数は以下同様に表されます。たとえば、5つの SDR がある場合、それらは SDR-0 から SDR-4 として識別されます。

図 17: データ収集階層のサポート



新しい SDR が作成されると、すべての数値が 1 ずつ増分され、最新の SDR には 0 の値が割り当てられます。レコードの合計数が設定された最大値を超えると、最も古い SDR が削除されます。

上記の例では、最大 SDR 数が 5 に達すると、SDR は SDR-0 から SDR-4 を継続し、ファイルのタイムスタンプはファイルが時間の経過とともに変化していることを示しています。

コレクション間の時間間隔は、指定されたスリープ期間に対して数分で異なる場合があります。これは、間隔がスケジュールされた収集が実行されるまでのアイドル時間を指定するためです。収集プロセスの実際のオーバーヘッドはスケジュールされた間隔に含まれていないため、コレクション間の時間の差にはこの非決定的時間が含まれます。



**重要** この動作を補うために短い間隔を使用することは推奨されません。これは、収集プロセスによって発生するオーバーヘッドが増えるだけで、最終的にシステム全体のパフォーマンスに影響するためです。スケジュールされたコレクション間のスリープ期間（アイドル時間）は、ユーザーによって回避されない「自己スロットリング」メカニズムの重要なコンポーネントです。

Exec モードの **show support collection** コマンドは、サポートデータコレクターに関する有用な情報を表示します。出力には、コレクタが最後に実行された時刻、実行にかかった時間、再実行がスケジュールされた時点、保存されている SDR の数、および使用されているストレージ容量に関する情報が含まれます。このコマンドの詳細については、[Execモードのコマンド \(423 ページ\)](#) を参照してください。

## SDR を使用した問題の診断

ユーザーは、2 つ以上のシーケンスを検査することで、SDR を比較できます。これらの SDR は、CLI 形式の出力表示にダンプされます。表示出力を比較すると、問題領域を示す傾向とパフォーマンス、または設定の違いが明らかになります。

特定のレコードセクションが問題のある特性を持つと確認されたら、それらのセクションに関連付けられている **CLI show** コマンドのみをモニターして比較し、問題領域をさらに分離する必要があります。また、個々の SDR は、システムでサポートされているプロトコルを介してリモートシステムに転送されるか、または現在の収集を後で分析するためのセットとして転送される場合があります。

## SDR CLI コマンド

収集されたサポートデータレコードを使用して、時系列でサポートデータを表示できます。デフォルトのリストと一連のセクションがシステムモニタリングに十分でない場合は、特定のサポートレコードを構成する一連の独自のレコードセクションコマンドを設定できます。



**重要** レコードセクションでサポートされている CLI 文字列のリスト (*show* コマンド) については、付録の「**SDR CLI コマンドの文字列**」を参照してください。また、このリストでは、デフォルトのレコードセクションとしてサポートされている CLI 文字列も確認できます。**show support collection definitions** コマンドを実行すると、同じリストを取得できます。



**重要** 1つのレコードセクションコマンドで最大 200 個の SDR CLI 文字列を入力できます。200 個を超える CLI 文字列を追加しようとすると、エラーメッセージが表示されます。また、解析するには複雑すぎるため、要求されたすべての CLI 文字列をシステムで解析できない場合にも、エラーメッセージが表示されることがあります。

SDR を設定したら、レコード収集間のスリープ間隔と、上書きされる前に保持する過去のレコードの数を設定します。デフォルトでは、この収集情報を設定スロと、収集メカニズムがアクティブになります (まだアクティブになっていない場合)。

1つ以上の収集間隔が経過すると、SDR データが分析に使用できるようになります。管理者は CLI コマンドを使用して SDR 情報を確認し、時間の経過にともなってデータがどのように変化したかに基づいて、根本原因の分析と傾向分析を実行できます。管理者は、Cisco TAC などによってリモートで分析されるように、SDR をシステムから転送することができます。

以下で説明する CLI コマンドの詳細については、『*Command Line Interface Reference*』を参照してください。

## 設定コマンド (グローバルコンフィギュレーションモード)

### support record

```
support record section section-name command "command-string" [ section
section-name command "command-string" ] ...
```

```
no support record [ all | section section_name ]
```

```
default support record [ all | section section_name ]
```

**support record section** コマンドは、support information output コマンドの特定のレコードセクションまたはレコードセクションのセットを設定します。レコードセクションが保存される順序は、CLI コマンドが入力された順序に関係なく、固定されています。

次に例を示します。

```
[local]host_name(config)# support record section show_context command "show
context"
```

**support record section** コマンドがユーザーによって明示的に設定されていない場合は、record section コマンドのデフォルトセットが使用されます。これらのデフォルトの record section コマンドは、**show configuration verbose** コマンドを実行すると表示されます。support record section コマンドが明示的に設定されている場合は、デフォルトのコマンドが置き換えられます。



**重要** レコードセクションでサポートされている CLI 文字列のリスト (*show* コマンド) については、付録の「**SDR CLI コマンドの文字列**」を参照してください。このリストでは、デフォルトのレコードセクションに含まれている CLI 文字列も確認できます。

**no support record** コマンドを実行すると、レコード定義の特定のセクションか、またはすべてのセクションが削除されます。**default support record** コマンドを指定した場合は、その指定されたレコードセクションのデフォルトのレコードセクションの定義が使用されます。キーワードの **all** または **section** が指定されていない場合は、すべてのレコードセクションの定義が削除されます。

## support collection

```
support collection [ sleep-duration [ hours h | minutes m ] ] [ max-records n ]
```

```
no support collection
```

```
default support collection
```

**support collection** コマンドは、サポート収集プロセスの変更や有効化を行います。support collection がすでに無効になっている場合、このコマンドは収集アクティビティを有効にします。support collection が現在有効になっている場合は、このコマンドを使用して、スリープ時間間隔や、収集して保存できる SDR の最大数を変更できます。

**sleep duration** キーワードは、サポートデータの収集間の時間間隔を指定します。時間または分単位で指定できます。デフォルトは1時間（60分）です。

**max-records** キーワードは、保存する SDR の数を 1～65535 の整数で指定します。この値を超えると、新しい SDR は最も古い SDR を上書きします。デフォルト値は 168 です。



---

**重要** SDR ファイルは `/hd-raid/support/records/` ディレクトリに保存されます。

---

次に例を示します。

```
[local]host_name(config)# support collection sleep-duration minute 30 max-records 50
```

SDR の収集を明示的に無効にするには、**no support collection** コマンドを使用します。record section コマンドが定義されていない場合は、サポートデータコレクターのメカニズムも実質的に無効化されます。

デフォルトのレコードセクションを使用してサポートデータコレクターを有効にするには、**default support collection** コマンドを使用します。

## Exec モードのコマンド

### show support record

```
show support record record-id [ to record-id ] [ section section_name ]
```

**show support record** コマンドを実行すると、SDR の収集が表示されます。SDR は、最も低いレコード ID から最も高いレコード ID までの順に表示されます。

各 SDR は、レコード ID と呼ばれる時間のインデックスによって識別されます。たとえば、最新のレコードは常に 0 (filename=sdr.0.gz) です。次に古いレコードは、レコード ID 1 (filename = sdr.1.gz) などです。

新しいレコードが収集されると、レコード ID 0 になります。前の最新のレコードの名前がレコード ID 1 などに変更されます。この表示には、収集タイムスタンプとともにレコード ID が含まれています。

レコード ID の変数は、単一の SDR を識別します。to キーワードは、SDR の範囲を表示するときにエンドポイントのレコード ID を指定します。

section キーワードは、レコードの特定のセクションが表示されます。

## delete support record

```
delete support record record-id [ to record-id ]
```

**delete support records** コマンドは、指定されたレコード ID を持つ SDR、または指定された範囲のレコード ID を持つすべての SDR を削除します。

## show support collection

```
show support collection [ definitions ]
```

**show support collection** コマンドは、SDC アクティビティに関する情報を表示します。このコマンドは、最後にスケジュールされた収集の開始時刻、最後にスケジュールされた収集の期間、収集がまだ進行中かどうかなどの情報を表示します。さらに、現在保存されている一連の SDR レコード ID、それぞれのタイムスタンプ、各 SDR のサイズを一覧表示します。

```
[local]host_name# show support collection
Record Collection Enabled : yes
Last Collection Start Time : Monday October 21 06:29:05 PDT 2013
Last Collection End Time   : Monday October 21 06:29:09 PDT 2013
Est. Collection Next Start : Monday October 21 07:29:13 PDT 2013 (40 minutes)
```

```
Support Data Records at /var/tmp/support-records/
  ID      Name      Size      Date/Time
  167    sdr.167.gz    42863    Monday October 21 04:40:00 PDT 2013
  166    sdr.166.gz    170425    Monday October 21 05:40:08 PDT 2013
total SDRs 2, total bytes 2132880, time span is last 1 day(s) 1 hour(s)
```

オプションの **definitions** キーワードを指定すると、デフォルトのサポートレコードセクションの定義のリストが表示されます。これは、有効なすべてのレコードセクションの定義のリストです。また、この表示にはレコードセクションがデフォルトで有効になっているか、無効になっているかも示されます。

```
[local]host_name# show support collection definitions
```

このコマンドの出力は、ユーザーが入力した順序に関係なく、レコードセクションが出力される順序を反映しています。詳細については、「*SDR CLI* のコマンド文字列」を参照してください。





## 付録 **A**

# エンジニアリングルール

この付録では、ネットワークの展開要件を満たすようにシステムを設定するためのエンジニアリングガイドラインについて説明します。



(注) この付録に記載されているエンジニアリングルールは、StarOSの最大キャパシティを反映しています。VMで実行しているVPCの実際の制限は、インスタンスに割り当てられたvCPUとvMemoryのキャパシティによって異なります。

- [CLIセッションルール \(425 ページ\)](#)
- [VPC インターフェイスとポートのルール \(426 ページ\)](#)
- [コンテキストルール \(427 ページ\)](#)
- [サブスクライバルール \(430 ページ\)](#)
- [サービスルール \(431 ページ\)](#)
- [Access Control List \(ACL; アクセスコントロールリスト\) のエンジニアリングルール \(431 ページ\)](#)
- [ECMP グループ \(432 ページ\)](#)
- [VPN スケーリングの要件 \(433 ページ\)](#)

## CLI セッションルール

使用可能なメモリの量に基づく CLI セッションのサポートは複数あります。内部 Resource Manager は、常に 6 つ以上の CLI セッションをサポートするのに十分なリソースを予約します。さらに、6 つのセッションのうち 1 つは、シリアルインターフェイス上の CLI セッションによる排他的使用のために予約されています。

十分なリソースが使用可能な場合は、事前予約された制限を超える追加の CLI セッションが許可されます。Resource Manager が事前に予約されているリソースを超えて CLI セッションのリソースを予約できない場合、管理権限を持つユーザーは、予約されているリソースがなくても、新しい CLI セッションを作成するように求められます。

# VPC インターフェイスとポートのルール

この項で説明するルールは、サブスクライバトラフィックのハイパーバイザを介して指定された vNIC イーサネットポートに関連しています。

## vNIC イーサネットポート

- すべての hypervisorassigned 割り当てられた論理インターフェイスに一意的な名前を付けて、同じコンテキスト内の他のインターフェイスとインターフェイスを区別します。
- 1つの仮想ポートは、そのポートの VLAN タグを設定するときに、ハイパーバイザを割り当てられた複数の論理インターフェイスをサポートできます。VLAN タグを使用すると、異なるコンテキストに存在する複数の論理インターフェイスに1つのポートをバインドできます。
- サブスクライバトラフィック用の各 vNIC ポートには、最大 1024 の VLAN タグ (VPC シャーシあたり最大 4000 の VLAN) を含めることができます。
- ハイパーバイザを割り当てられたすべての論理インターフェイスには、有効な IP アドレスとサブネットが必要です。
  - ネットワークでマルチホーミングがサポートされている場合は、すべての論理インターフェイスに1つのプライマリ IP アドレスと最大 16 個のセカンダリ IP アドレスを割り当てることができます。
- 1つの StarOS 論理 (名前付き) インターフェイスをコンテキストごとに設定します。この名前付きインターフェイスは、最大 512 の ethernet+ppp+tunnel インターフェイスを持つことができます。
- 異なる StarOS コンテキストは、同じ論理 (名前付き) インターフェイスを共有できます。
- 最大 256 のアクセスコントロールリスト (ACL) ルールを StarOS 論理インターフェイスに適用できます。
- StarOS では、すべてのポートが <slot>/<port> によって識別されます。

## パケットデータネットワーク (PDN) インターフェイスのルール

パケットデータネットワーク (PDN) へのインターフェイスには、次のエンジニアリングルールが適用されます。

- 出力コンテキスト内の PDN インターフェイスを容易にするために使用される論理インターフェイスを設定します。
- デフォルトでは、出力コンテキスト内で1つのインターフェイスを使用し、PDN インターフェイスを容易にします。

- スタティックルートまたはダイナミックルーティングプロトコルを使用して、出力コンテキスト内に複数のインターフェイスを設定できます。
- また、ネクストホップのデフォルトゲートウェイを設定することもできます。

## コンテキストルール

- シャーシごとに最大 63 のコンテキストを設定できます。MIO カードでデマルチプレクサ機能を有効にすると、コンテキストの最大数が 10 に低減します。
- コンテキストごとのインターフェイス
  - Demux MIO/UMIO/MIO2 機能を有効にすると、最大 64 のインターフェイスを 1 つのコンテキスト内に設定できます。
  - 512 イーサネット + PPP + トンネルインターフェイス
  - 32 ipv6ip トンネルインターフェイス
  - 511 GRE トンネル (シャーシあたり 2,048 GRE トンネル)
  - 256 ループバック インターフェイス
- IP アドレスと IP アドレスプール
  - 最大 2,000 の IPv4 アドレスプールを 1 つのコンテキスト内に設定できます。
  - 最大 256 の IPv6 プールを 1 つのコンテキスト内に設定できます。
  - 合計で 5,000 の IPv4 アドレスと IPv6 アドレスをシャーシごとに設定できます。
  - 各コンテキストは、最大 3,200 万のスタティック IP プールアドレスをサポートします。シャーシごとにあたり合計で最大 9,600 万のスタティック IP プールアドレスを設定できます。各スタティック IP プールには、最大 50 万のアドレスを含めることができます。
  - 各コンテキストは、最大 1,600 万のダイナミック IP プールアドレスをサポートします。シャーシあたり合計で最大 3,200 万のダイナミック IP プールアドレスを設定できます。各ダイナミック IP プールには、最大 50 万のアドレスを含めることができます。



### 重要

コンテキストとシャーシあたりでサポートされる IP プールの実数の数は、使用しているアドレスの数とサブネット化の方法によって異なります。



**重要** プール内の各アドレスには、約 60 バイトのメモリが必要です。ただし、必要なメモリの量は、プールタイプやホールドタイマーの使用率などのさまざまな要因によって異なります。そのため、設定するアドレスの数とインストールされているアプリケーションカードの数に応じて、プール数を制限して使用可能なメモリを節約する必要がある場合があります。

- 同時サブスクリバセッションの最大数は、サポートされているサービスのインストール済みのキャパシティライセンスによって制御されます。
- コンテキストあたりのスタティックアドレス解決プロトコル (ARP) エントリの最大数は 128 です。
- コンテキストあたりのドメインの最大数は 2,048 です。
- 同じコンテキスト内に設定された ASN-GW サービスは、相互に通信できません。
- ルート
  - コンテキストあたりの最大 1,200 のスタティックルート (シャージあたり 48,000)。
  - コンテキストあたり 6,000 のプールルート (シャージあたり 6,000)
  - コンテキストあたり 24,000 のプール (シャージあたり 24,000) の明示的ホストルート
  - コンテキストあたり 64 のルートマップ
- BGP
  - コンテキストあたり 64,000 の BGP プレフィックス (シャージあたり 64,000) を学習/アドバタイズできます。
  - コンテキストあたり 64 の EBGP ピアの設定が可能 (シャージあたり 512)
  - コンテキストあたり 16 の IBGP ピア
  - シャージ間セッションリカバリ (ICSR) のサポートのサポートにコンテキストあたり 512 の BGP/AAA モニター
- OSPF
  - シャージあたり 200 の OSPF ネイバー
  - コンテキストあたり 10,000 の OSPF ルート (シャージあたり 64,000)
- MPLS
  - リリース 21.6 まで
    - コンテキストあたり 16 のラベル配布プロトコル (LDP) セッション

- コンテキストあたり最大 8,000 の着信ラベルマップ (ILM) エントリ (シャーシあたり 48,000)
- 128,000 のネクストホップラベル転送エントリ (NHLFE) と発生する可能性がある 64,000 のプレフィックスのテーブルサイズを組み合わせます。
  - コンテキストあたり 1,000 の転送等価クラス (FEC) エントリ (シャーシあたり 4,000) : 32 のパスを含む
  - コンテキストあたり 2,000 の転送等価クラス (FEC) エントリ (シャーシあたり 8,000) : 16 のパスを含む
  - コンテキストあたり 16,000 の転送等価クラス (FEC) エントリ (シャーシあたり 64,000) : 2 つのパスを含む
  - コンテキストあたり 64,000 の転送等価クラス (FEC) エントリ (シャーシあたり 64,000) : 1 つのパスを含む
- リリース 21.7 以降
  - コンテキストあたり 16 のラベル配布プロトコル (LDP) セッション
  - コンテキストあたり最大 8,000 の着信ラベルマップ (ILM) エントリ (シャーシあたり 48,000)
  - 256,000 のネクストホップラベル転送エントリ (NHLFE) と発生する可能性がある 64,000 のプレフィックスのテーブルサイズを合算します。
    - コンテキストあたり 1,000 の転送等価クラス (FEC) エントリ (シャーシあたり 4,000) : 64 のパスを含む
    - コンテキストあたり 2,000 の転送等価クラス (FEC) エントリ (シャーシあたり 8,000) : 32 のパスを含む
    - コンテキストあたり 32,000 の転送等価クラス (FEC) エントリ (シャーシあたり 64,000) : 2 つのパスを含む
    - コンテキストあたり 64,000 の転送等価クラス (FEC) エントリ (シャーシあたり 64,000) : 1 つのパスを含む
- VRF
  - コンテキストあたり 300 の Virtual Route Forwarding (VRF) テーブル (シャーシあたり 2,048 の VRF) (MIO カードで有効になっている demux 機能を使用した場合、コンテキストあたり 256 の VRF)
  - APN の制限はシャーシあたり 2,048 です。VRF の制限と APN の制限は同一である必要があります。
  - 64,000 の IP ルート
- NEMO (ネットワークモビリティ)

- シャーシあたり 512,000 のプレフィックス/フレーム化されたルートと、MR（モバイルルータ）あたり最大 16 の動的に学習されたプレフィックス
- デフォルトの AAA サーバークラスの場合にはコンテキストあたり 128 の AAA サーバークラスは、アカウント、認証、課金サーバークラス、またはそれらの組み合わせとして設定できます。
- 次の制限事項を踏まえて、コンテキストあたり最大 800 の AAA サーバークラスを設定できます。
  - AAA サーバークラスあたり 128 のサーバークラス（アカウント、認証、課金サーバークラス、またはその組み合わせ）
  - AAA サーバークラスモードあたり 1,600 のサーバークラス（アカウント、認証、課金サーバークラス、またはその組み合わせ）
  - コンテキストあたり 800 の NAS-IP アドレス/NAS 識別子（サーバークラスあたりプライマリ 1、セカンダリ 1）
- GTPP アカウント用としてコンテキストあたり最大 12 の課金ゲートウェイ機能（CGF）を設定できます。
- コンテキストあたり最大 16 の Bidirectional Forwarding Detection（BFD）セッション（シャーシあたり 64）



**重要** 製品固有の動作制限の詳細については、製品管理ガイドの「エンジニアリングルール」を参照してください。

## サブスクライバルール

システム内に設定されているサブスクライバには、次のエンジニアリングルールが適用されません。

- コンテキストごとに最大 2048 のローカルサブスクライバを設定します。
- 各ローカルサブスクライバの属性を設定できます。
- コンテキストが確立されると、システムはコンテキストごとにデフォルトのサブスクライバを作成します。各デフォルトサブスクライバの属性を設定します。AAA ベースのサブスクライバが認証応答メッセージに属性がない場合、サブスクライバが認証されたコンテキスト内のデフォルトのサブスクライバ属性が使用されます。



**重要** ローカル認証（ローカルサブスクライバの場合）が実行される場合、デフォルトは使用されません。

- AAA レルム（コンテキスト内で設定されたドメインエイリアス）ごとにデフォルトのサブスクライバテンプレートを設定します。
- PDSN、FA、ASNGW、またはHA サービスごとにデフォルトのサブスクライバテンプレートを設定します。
- AAA 認証サブスクライバの場合、属性の設定に使用するローカルサブスクライバテンプレートの選択は、次の順序で行われます。
  - ユーザー名（NAI）が任意のローカルドメイン名と一致し、ドメイン名にローカルサブスクライバ名が設定されている場合、そのローカルサブスクライバテンプレートが使用されます。
  - 最初のケースが失敗し、提供サービスにデフォルトのユーザー名が設定されている場合は、そのサブスクライバテンプレートが使用されます。
  - 最初の2つのケースが失敗した場合は、AAA コンテキストのデフォルトのサブスクライバテンプレートが使用されます。

## サービスルール

システム内に設定されているサービスには、次のエンジニアリングルールが適用されます。

- システムごとに（タイプに関係なく）最大 256 のサービスを設定します。



**注意** 多数のサービスによって管理の複雑度が大幅に増大し、システム全体のパフォーマンスに影響を与える可能性があります。したがって、アプリケーションが絶対に必要としない限り、多数のサービスを設定する必要はありません。詳細については、シスコのサービス担当者にお問い合わせください。

- テーブルあたりおよびシャーシあたりのエントリの合計数は、256 に制限されます。
- 同じシステム上の異なるコンテキストで設定されたものと同じサービス名を使用できますが、これは適切な方法ではありません。同じ名前のサービスを使用すると、混乱が生じ、問題のトラブルシューティングが困難になる可能性があります。また、**show** コマンドの出力を理解することが困難になります。

## Access Control List（ACL; アクセスコントロールリスト）のエンジニアリングルール

アクセスコントロールリストには、次のルールが適用されます。

- ACL ごとルールの最大数は 128 です。
- ポートごとに適用される ACL ルールの最大数は 128 です。
- コンテキストごとに適用される ACL ルールの最大数は 1,024 です。
- IPSec ポリシーごとの ACL ルールの最大数は 1 です。
- コンテキストごとの IPSec ACL ルールの最大数は 1,024 です。
- 暗号マップごとの IPSec ACL ルールの最大数は 8 です。
- コンテキストごとに設定できる ACL の最大数は、各 ACL 内で許可されるルールの数によって制限されます。各 ACL にルールの最大数 (128) が含まれている場合、コンテキストあたりの ACL の最大数は 8 (128 X 8 ACLs = 1,024 ACL rules per context) です。
- IP アクセスグループに適用される ACL の最大数は、ポートまたはコンテキストに設定されているかどうかにかかわらず、1 です。インターフェイスまたはコンテキストに適用できる IP アクセスグループの最大数は 16 であるため、次の計算が適用されます。
  - 各インターフェイス/ポートについて : ACL ごとの 8 つのルールと 16 の IP アクセスグループの乗数 = 128 (ポートごとの ACL ルールの制限)
  - 各コンテキストについて : ACL ごとの 64 ルールと 16 IP アクセスグループの乗数 = 1,024 (コンテキストごとの ACL ルールの制限)

## ECMP グループ

等コストマルチパス (ECMP) グループの最大数は次のとおりです。

- 21.4 よりも前のリリースでは、StarOS は最大 64,000 グループをサポートしています。
- 21.4 以降のリリースでは、StarOS は最大 32,000 グループをサポートしています。



- (注)
- *max\_num* は 1 ~ 10 の整数です。
    - リリース 21.4x
      - QVPC-DI: 64
      - QVPC-SI: 64
      - ASR 5500: 24
  - [設定の確認と保存 \(111 ページ\)](#) の章の説明に従って、設定を保存します。



# VPN スケーリングの要件

特定のリリースでは、次の VPN スケーリング番号がサポートされています。

パラメータ	スケーリング番号 (リリース 12.x、14.x)	スケーリング番号 (リリース 15.x、16.x)	スケーリング番号 (リリース 17.x、18.x、19.x、20.x +)
BFD セッション	コンテキストあたり 16 シャーシあたり 64	コンテキストあたり 16 シャーシあたり 64	コンテキストあたり 16 シャーシあたり 64
コンテキストレベルの ACL	コンテキストあたり 256	コンテキストあたり 256	コンテキストあたり 256
ダイナミックプールアドレス	コンテキストあたり 1,600 万 シャーシあたり 3,200 万	コンテキストあたり 1,600 万 シャーシあたり 3,200 万	コンテキストあたり 1,600 万 シャーシあたり 3,200 万
コンテキストごとの IPv4 プール	コンテキストあたり 2,000 シャーシあたり 5,000 (IPv4 と IPv6 の組み合わせ)	コンテキストあたり 2,000 シャーシあたり 5,000 (IPv4 と IPv6 の組み合わせ)	コンテキストあたり 2,000 シャーシあたり 5,000 (IPv4 と IPv6 の組み合わせ)
コンテキストごとの IPv6 プール	各コンテキストに 32 シャーシあたり 5,000 (IPv4 と IPv6 の組み合わせ)	コンテキストあたり 256 シャーシあたり 5,000 (IPv4 と IPv6 の組み合わせ)	コンテキストあたり 256 シャーシあたり 5,000 (IPv4 と IPv6 の組み合わせ)
BGP プレフィックスの数	コンテキストあたり 16,000 シャーシあたり 64,000	コンテキストあたり 32,000 シャーシあたり 64,000	コンテキストあたり 64,000 シャーシあたり 64,000
コンテキストの数	63 (ただし、コンテキストが 32 を超えると、PSC の移行は正常に機能しません)	63 (ただし、コンテキストが 32 を超えると、PSC の移行は正常に機能しません)  この項の最後にある「MIO カード上のデマルチプレクサ」に関する情報に注意してください。	63 (ただし、コンテキストが 32 を超えると、PSC の移行は正常に機能しません)  この項の最後にある「MIO カード上のデマルチプレクサ」に関する情報に注意してください。

パラメータ	スケーリング番号 (リリース 12.x、14.x)	スケーリング番号 (リリース 15.x、16.x)	スケーリング番号 (リリース 17.x、18.x、19.x、20.x+)
動的に学習したMRあたりのプレフィックスの数	8	16	16
EBGP ピアの数	コンテキストあたり 64 シャーシあたり 512	コンテキストあたり 64 シャーシあたり 512	コンテキストあたり 64 シャーシあたり 512
FEC エントリの数	コンテキストあたり 8,000 ラベル  シャーシあたり 48,000	コンテキストあたり 8,000 ラベル  シャーシあたり 48,000	コンテキストあたり 8,000 ラベル  シャーシあたり 48,000
IBGP ピアの数	コンテキストあたり 16	コンテキストあたり 16	コンテキストあたり 16
ILM エントリの数	コンテキストあたり 8,000 ラベル  シャーシあたり 48,000	コンテキストあたり 8,000 ラベル  シャーシあたり 48,000	コンテキストあたり 8,000 ラベル  シャーシあたり 48,000
インターフェイスの数	コンテキストあたり 512イーサネット+ppp +トンネルインター フェイス  コンテキストあたり 32 IPv6 IP トンネルイン ターフェイス  コンテキストあたり最大 511 GRE トンネルお よびシャーシあたり最大 2,048 GRE トンネル  コンテキストあたり 256 ループバック イン ターフェイス	コンテキストあたり 512イーサネット+ppp +トンネルインター フェイス  コンテキストあたり 32 IPv6 IP トンネルイン ターフェイス  コンテキストあたり最大 511 GRE トンネルお よびシャーシあたり最大 2,048 GRE トンネル  コンテキストあたり 256 ループバック イン ターフェイス  この項の最後にある 「MIOカード上のデマ ルチプレクサ」に関す る情報に注意してくだ さい。	コンテキストあたり 512イーサネット+ppp +トンネルインター フェイス  コンテキストあたり 32 IPv6 IP トンネルイン ターフェイス  コンテキストあたり最大 511 GRE トンネルお よびシャーシあたり最大 2,048 GRE トンネル  コンテキストあたり 256 ループバック イン ターフェイス  この項の最後にある 「MIOカード上のデマ ルチプレクサ」に関す る情報に注意してくだ さい。
LDP セッションの数	コンテキストあたり 16	コンテキストあたり 16	コンテキストあたり 16

パラメータ	スケーリング番号 (リリース 12.x、14.x)	スケーリング番号 (リリース 15.x、16.x)	スケーリング番号 (リリース 17.x、18.x、19.x、20.x +)
NEMO プレフィックス/フレームルートの数	シャーシあたり 256,000	シャーシあたり 512,000	シャーシあたり 512,000
OSPF ネイバーの数	シャーシあたり最大 200	シャーシあたり最大 200	シャーシあたり最大 200
OSPF ルートの数	コンテキストあたり最大 10,000 シャーシあたり 64,000	コンテキストあたり最大 10,000 シャーシあたり 64,000	コンテキストあたり最大 10,000 シャーシあたり 64,000
プールの明示的なホストルートの数	コンテキストあたり 5,000 (シャーシあたり 6,000)	コンテキストあたり 5,000 (シャーシあたり 6,000)	17.x および 18.[1234] でコンテキストあたり 5,000 (シャーシあたり 6,000)  18.5以降でコンテキスト あたり 24,000 (シャーシあたり 24,000)
プールルートの数	コンテキストあたり 6,000 (シャーシあたり 6,000)	コンテキストあたり 6,000 (シャーシあたり 6,000)	コンテキストあたり 6,000 (シャーシあたり 6,000)
ルートの数 (フレームルートを除く)	コンテキストあたり 64,000	コンテキストあたり 64,000	コンテキストあたり 64,000
インターフェイスあたりのセカンダリアドレスの数	16	16	16
スタティックルートの数	コンテキストあたり 1,200	コンテキストあたり 1,200	コンテキストあたり 1,200
VLAN の数	シャーシあたり 4,000	シャーシあたり 4,000	シャーシあたり 4,000

パラメータ	スケーリング番号 (リリース 12.x、14.x)	スケーリング番号 (リリース 15.x、16.x)	スケーリング番号 (リリース 17.x、18.x、19.x、20.x+)
VRF の数	<p>コンテキストあたり 250</p> <p>シャーシあたり 2,048</p> <p>APN の制限は 1024/シャーシであり、VRF の制限と一致しません。</p>	<p>コンテキストあたり 300</p> <p>シャーシあたり 2,048</p> <p>(注)</p> <ul style="list-style-type: none"> <li>• VRF の制限と APN の制限は同一であると想定しています。</li> <li>• この項の最後にある「MIO カード上のデマルチプレクサ」の項に注意してください。</li> </ul>	<p>コンテキストあたり 300</p> <p>シャーシあたり 2,048</p> <p>注：VRF の制限と APN の制限は同一であると想定しています。</p> <p>この項の最後にある「MIO カード上のデマルチプレクサ」の項に注意してください。</p>
ルートの数 (フレームルートを含むすべての種類のルート)	コンテキストあたり 64,000	コンテキストあたり 64,000	コンテキストあたり 64,000

パラメータ	スケーリング番号 (リリース 12.x、14.x)	スケーリング番号 (リリース 15.x、16.x)	スケーリング番号 (リリース 17.x、18.x、19.x、20.x +)
ルート マップ	コンテキストあたり 64	コンテキストあたり 64	コンテキストあたり 64
スタティックプールアドレス	コンテキストあたり 3200 万 シャーシあたり 9600 万	コンテキストあたり 3200 万 シャーシあたり 9600 万	コンテキストあたり 3200 万 シャーシあたり 9600 万

### MIO カードでの Demux

MIO カードで Demux を有効にすると、VPN リソースは MIO カードでコントローラプロセスと結合されるため、すべての VPN タスクで使用可能なリソースが減少します。これにより、MIO カードが demux-enabled の場合に、制限の一部（前の項で説明）が削減されます。





## 付録 **B**

# StarOS のタスク

この付録では、ASR 5500 と仮想プラットフォーム上の StarOS で実行されているシステムとサブシステムのタスクについて説明します。



**重要** この付録は、すべての StarOS タスクの包括的なリストではありません。StarOS 内のプライマリタスクとサブシステムの一般的な説明だけを示します。

内容は次のとおりです。

- [概要 \(439 ページ\)](#)
- [プライマリ タスク サブシステム \(440 ページ\)](#)
- [コントローラおよびマネージャ \(442 ページ\)](#)
- [サブシステムのタスク \(442 ページ\)](#)

## 概要

冗長性、拡張性、および堅牢なコール処理のために、StarOS は特定の機能を実行する一連のタスクをサポートします。これらのタスクは、必要に応じて、制御信号とデータ信号を共有するために相互に通信します。その結果、複数のタスクにまたがってプロセスを分散させることができるため、特定のタスクの全体的なワークロードが軽減され、システムのパフォーマンスが向上します。この分散型設計は、障害によるプロセスまたはセッションへの影響を著しく軽減する障害抑制を提供します。

Exec モードの **show task** コマンドは、StarOS 内の実行中のプロセスのスナップショットを表示します。このコマンドの詳細については、『*Command Line Interface Reference*』および『*Statistics and Counters Reference*』を参照してください。

次の項では、StarOS によって実装されるプライマリタスクについて説明します。

- [プライマリ タスク サブシステム \(440 ページ\)](#)
- [コントローラおよびマネージャ \(442 ページ\)](#)

# プライマリ タスク サブシステム

CPU上で実行される個々のタスクは、サブシステムに分割されます。次に、コールセッション処理を担当する主要なサブシステムのリストを示します。

- **システム開始タスク (SIT)** : このサブシステムはタスクを開始し、システムを初期化します。これには、システムのスタートアップ時間 (静的タスク) に一連の初期タスクを開始すること、および任意の時間にオンデマンドで個々のタスクを開始すること (動的タスク) が含まれます。
- **高可用性タスク (HAT)** : リカバリ制御タスク (RCT) のサブシステムを使用すると、HAT サブシステムはシステムの動作状態を維持します。HAT は、システムのさまざまなソフトウェアコンポーネントおよびハードウェアコンポーネントをモニターします。異常な動作 (別のタスクの予期しない終了など) がある場合、HAT サブシステムは適切なアクションを実行します。たとえば、イベントを RCT サブシステムにトリガーして修正措置を実行したり、ステータスをレポートしたりします。HAT タスクの主な機能は、サービスへの影響を最小限に抑えることです。
- **リカバリ制御タスク (RCT)** : このサブシステムは、システムで発生した障害に対してリカバリアクションを実行します。RCT サブシステムは、HAT サブシステムから信号を受信し (NPUサブシステムからの場合もある)、必要なリカバリアクションを決定します。
- **共有設定タスク (SCT)** : このサブシステムは、システム設定パラメータの設定、取得、および通知を受信する機能を提供します。SCTは主に、システム上で実行されているアプリケーションの設定データの保存を担当します。

SCTサブシステムは、アクティブな管理カード上でのみ動作し、そこに含まれている情報をスタンバイ管理カード上の SCT サブシステムと同期します。

- **リソース管理 (RM)** : このサブシステムは、スタートアップ時にシステムタスクごとに、CPUの負荷とメモリなどのリソースを割り当てます。RMサブシステムは、リソースの使用状況をモニターして、割り当てが指定どおりであることを確認します。また、RMは、すべてのセッションをモニターし、セッションコントローラと通信して、キャパシティライセンスの制限を強制的に適用します。
- **バーチャル プライベート ネットワーク (VPN)** : このサブシステムは、システム内のすべての VPN 関連のエンティティの管理面および運用面を管理します。VPN サブシステムによって実行される機能には、次のようなものがあります。
  - 個別の VPN コンテキストの作成
  - VPN コンテキスト内の IP サービスの開始
  - IP プールとサブスクライバ IP アドレスを管理し、VPN コンテキスト内で IP フロー情報を配布します。

StarOS 内のすべての IP 操作は、特定の VPN コンテキスト内で実行されます。一般に、パケットは異なる VPN コンテキスト間で転送されません。現在の唯一の例外は、セッションサブシステムです。



- **ネットワーク処理ユニット**（ASR 5500 では npusim、および VPC-DI と VPC-SI では iftask または knpusim）<sup>1</sup>：このサブシステムは次の役割を担います。
  - データベースを使用したアドレスおよびポート番号と接続先タスクの照合によるデータフレームの高速パス転送
  - さまざまな物理インターフェイスとの間でのユーザーデータフレームの送受信
  - IP 転送の決定（ユニキャストとマルチキャストの両方）
  - インターフェイスごとのパケットのフィルタリング
  - トラフィック管理とトラフィック エンジニアリング
  - パケット処理の CPU との間でのユーザーデータフレームの受け渡し
  - データリンク層とネットワーク層のヘッダーの変更、追加、または除去
  - チェックサムの再計算
  - 統計情報の維持
  - 外部イーサネット インターフェイスの管理
- **カード/スロット/ポート（CSP）**：カードが挿入、ロック、ロック解除、削除、シャットダウン、または移行されたときに発生するイベントを調整します。また、CSP は自動検出を実行し、新しく挿入されたインターフェイスカードのポートを設定します。これにより、インターフェイスカードがパケット処理カードにマッピングされる方法が決定されます。

CSP サブシステムは、アクティブな管理カード上でのみ動作し、そこに含まれている情報をスタンバイ管理カード上の SCT サブシステムと同期します。これは、SIT サブシステムによって開始され、HAT サブシステムによってモニターされます。
- **セッションマネージャ（SM）**：モバイルサブスクライバのパケット指向データセッションフローのハイタッチ処理を実行します。ハイタッチユーザーのデータ処理は、次のもので構成されます。
  - ペイロード変換
  - フィルタリングとスケジューリング
  - 統計情報の収集
  - ポリシング

<sup>1</sup> の knpusim は、Intel DPDK がサポートされていない設定の VPC VM で iftask の代わりに実行されます。

## コントローラおよびマネージャ

プライマリサブシステムの多くは、コントローラと呼ばれるコントローラタスクと、マネージャと呼ばれる下位のタスクで構成されています。

コントローラはいくつかの目的に役立ちます。

- マネージャの状態をモニターし、同じサブシステム内のマネージャ間での通信を可能にします。
- 他のサブシステムのコントローラと通信できるため、サブシステム間の通信を可能にします。
- 管理を容易にするために、ソフトウェアの分散型の特性をユーザーからマスクします。

マネージャはリソースとリソース間のマッピングを管理します。さらに、一部のマネージャはコール処理を直接行います。

クリティカル、コントローラ、またはマネージャのタスク、あるいはそれらすべてのタスクで構成されるプライマリサブシステムの詳細については、[サブシステムのタスク \(442 ページ\)](#) を参照してください。

## サブシステムのタスク

次のサブセクションでは、さまざまなサブシステムの StarOS タスクについて簡単に説明します。

- [システム開始サブシステム \(443 ページ\)](#)
- [高可用性サブシステム \(444 ページ\)](#)
- [Resource Manager サブシステム \(445 ページ\)](#)
- [仮想プライベート ネットワーク サブシステム \(445 ページ\)](#)
- [ネットワーク処理ユニットサブシステム \(448 ページ\)](#)
- [セッションサブシステム \(449 ページ\)](#)
- [プラットフォームプロセス \(463 ページ\)](#)
- [管理プロセス \(467 ページ\)](#)

## システム開始サブシステム

表 30: システム開始サブシステムのタスク

タスク	説明	機能
SITMAIN	システム開始タスク：メイン	システムの起動時に開始されます。
		他の SIT コンポーネントの起動設定を読み取り、提供します。
		SITREAP サブ機能を起動します。
		CPU の状態情報を維持します。
SITPARENT	SIT の親サブ機能	管理カードをアクティブモードまたはスタンバイモードのいずれかで起動します。
		HAT タスクにタスクを登録します。
		CPU の起動完了の CSP タスクを通知します。
		スタンバイモードでパケット処理カードを起動します。
SITREAP	SIT REAP サブ機能	必要に応じてタスクをシャットダウンします。

## 高可用性サブシステム

表 31: 高可用性サブシステムのタスク

タスク	説明	機能
hatcpu	高可用性タスクの CPU	CPU のハードウェア機能に基づいて、デバイスの初期化と制御機能を実行します。
		CPU 上のタスクの損失を hatsystem サブ機能に報告します。
		パケット処理カードの LED を制御します。
		パケット処理カードの専用ハードウェアを初期化し、モニターします。(ASR 5500 のみ)
		CPU モニタリング情報を定期的に収集し、マスター hatcpu サブ機能に報告します。
		CPU 上のタスクの損失を master hatcpu サブ機能に報告します。
		CPU のハードウェア機能により、デバイスの初期化と制御機能を実行します。
		CPU 上のタスクの損失を hatsystem サブ機能に報告します。
		管理カードの LED を制御します。(ASR 5500 のみ)
		管理カードの専用ハードウェアを初期化し、モニターします。(ASR 5500 のみ)
hatsystem	高可用性タスクのシステムコントローラ	システム内のすべての HAT サブ機能のタスクを制御します。これは、システムの起動時に開始されます。
		システムコンポーネント (ギガビットイーサネットスイッチおよびスイッチファブリックなど) を初期化します。
		状態変更のためにファンなどのシステムコンポーネントをモニターします。
		障害検出の発生時に冗長性のアクションをトリガーします。
		冗長管理カードの HAT サブシステムは、アクティブな管理カード上の HAT サブシステムをミラーリングします。

## Resource Manager サブシステム

表 32: Resource Manager (RM) サブシステムのタスク

タスク	説明	機能
rmctrl	Resource Manager コントローラ	StarOS の起動時に sitparent タスクによって開始され、障害がないか HAT によってモニターされます。
		CPU やメモリなどのリソースを初期化します。
		CSP サブシステムから更新されたカードステータスを要求し、システムのカードテーブルを更新します。
		すべての rmctrls と通信して、最新のリソースデータのセットを要求します。
rmmgr	Resource Manager マネージャ	sitparent タスクによって開始され、障害の発生を HAT によってモニターされます。
		ローカルリソースのデータとローカルリソースのスクラッチスペースを初期化します。
		ローカル CPU 上の SIT タスクと通信して、タスクテーブル全体と、各タスクに関連付けられているリソースを取得します。
		各タスクの現在のリソース使用率を収集します。
		リソースデータを rmctrl タスクに送信します。

## 仮想プライベート ネットワーク サブシステム

表 33: 仮想プライベート ネットワーク (VPN) サブシステムのタスク

タスク	説明	機能
vpnctrl	VPN コントローラ	システムの起動時に作成されます。
		各コンテキストの VPN マネージャを起動します。
		コンテキストに追加または変更があった場合に、セッションコントローラ タスクに通知します。一度に動作するのは、1つのセッションコントローラだけです。
		コンテキスト固有の動作情報を適切な VPN マネージャにルーティングします。
		VPN マネージャのリカバリを実行し、すべての VPN 関連の設定情報を SCT に保存します。

タスク	説明	機能
vpnmgr	VPN マネージャ	設定されたコンテキストごとに VPN コントローラによって開始されます（ローカルコンテキストには常に1つ存在します）。
		IP アドレスプールとサブスクライバ IP アドレス管理を実行します。
		次のものを含みますが、これに限定されないすべてのコンテキスト固有の操作を実行します。UCM サービス、IP インターフェイス、Address Resolution Protocol (ARP)、IP アドレスプール管理、低速パス転送、NPU フロー、ポート Access Control List (ACL; アクセスコントロールリスト)、およびロギング。
		各コンテキストの IP インターフェイスアドレス情報をセッションコントローラに提供します。
bgp	ボーダー ゲートウェイ プロトコル	BGP ルーティングプロトコル ( <b>router bgp</b> コンテキスト コンフィギュレーション モードの CLI コマンド) を有効にしたコンテキストごとに、VPN マネージャによって作成されます。
		BGP プロトコルを介してルーティング情報の学習と再配布を担っています。
		BGP ピアリング接続を維持します。
		定義されている任意の BGP ルーティングポリシーを適用します。
ospf	Open Shortest Path First	OSPF ルーティングプロトコル ( <b>router ospf</b> コンテキスト コンフィギュレーション モードの CLI コマンド) を有効にしたコンテキストごとに、VPN マネージャによって作成されます。
		OSPF プロトコルを介してルーティング情報の学習と再配布を担っています。
		OSPF ネイバー関係を維持します。
		LSA データベースを維持します。
		SPF 計算を実行します。
		定義されている任意の OSPF ルーティングポリシーを適用します。

タスク	説明	機能
ospfv3	Open Shortest Path First	OSPFv3 ルーティングプロトコル ( <b>router ospfv3</b> コンテキスト コンフィギュレーションモードの CLI コマンド) を有効にしたコンテキストごとに、VPN マネージャによって作成されます。
		OSPFv3 プロトコルを介してルーティング情報の学習と再配布を担っています。
		OSPFv3 ネイバー関係を維持します。
		LSA データベースを維持します。
		OSPFv3 SPF 計算を実行します。
		定義されている任意の OSPFv3 ルーティングポリシーを適用します。
rip	ルーティング情報プロトコル	RIP ルーティングプロトコル ( <b>router rip</b> コンテキスト コンフィギュレーションモードの CLI コマンド) を有効にしたコンテキストごとに、VPN マネージャによって作成されます。
		RIP プロトコルを介してルーティング情報の学習と再配布を担っています。
		RIP データベースを維持します。
		定期的に RIP 更新メッセージを送信します。
		定義されている任意の RIP ルーティングポリシーを適用します。
zebos	L2 スイッチングおよび L3 スイッチング	コンテキストごとに VPN マネージャによって作成されます。
		コンテキストのルーティングテーブル (RIB および FIB) を維持します。
		スタティックルーティングを実行します。
		ルーティングとインターフェイス更新のためのカーネルへのインターフェイス。
		ルーティング情報をダイナミック ルーティングプロトコルに再配布します。
		ネクストホップの到達可能性を計算します。

## ネットワーク処理ユニットサブシステム

表 34: ネットワーク処理ユニット (NPU) サブシステムのタスク

タスク	説明	機能
iftask	内部フォワーダータスク (Intel DPDK) (VPC-DI、VPC-SI)	StarOS の起動時に作成されます。
		CSP タスクにポート設定サービスを提供します。
		VPN マネージャにインターフェイスバインディングサービスと転送サービスを提供します。
		セッションマネージャおよび AAA マネージャのタスクにフロー挿入サービスと削除サービスを提供します。
knpusim	カーネルベースの NPU シミュレータ (VPC-DI、VPC-SI)	StarOS の起動時に作成されます。
		CSP タスクにポート設定サービスを提供します。
		VPN マネージャにインターフェイスバインディングサービスと転送サービスを提供します。
		セッションマネージャおよび AAA マネージャのタスクにフロー挿入サービスと削除サービスを提供します。
		NPU コントローラにリカバリサービスを提供します。
npuctrl	NPU コントローラ	StarOS の起動時に作成されます。システムでは、常に 1 つの NPU コントローラのみが動作します。
		システム内の NPU マネージャの状態をモニターします。
		NPU マネージャがクラッシュしたときに通知を受信するように登録します。
		リカバリ動作を制御します。
		NPU マネージャの状態に関連する CLI コマンドの一元的な場所を提供します。
npumgr	NPU マネージャ	取り付けられ、起動しているすべてのパケット処理カードに対して作成されます。
		CSP タスクにポート設定サービスを提供します。
		VPN マネージャにインターフェイスバインディングサービスと転送サービスを提供します。
		セッションマネージャおよび AAA マネージャのタスクにフロー挿入サービスと削除サービスを提供します。
		NPU コントローラにリカバリサービスを提供します。



タスク	説明	機能
npusim	NPU シミュレータ (ASR 5500)	取り付けられ、起動しているすべての DPC に対して作成されます。
		CSP タスクにポート設定サービスを提供します。
		VPN マネージャにインターフェイスバインディングサービスと転送サービスを提供します。
		セッションマネージャおよび AAA マネージャのタスクにフロー挿入サービスと削除サービスを提供します。
		NPU コントローラにリカバリサービスを提供します。

## セッションサブシステム

表 35: セッションサブシステムのタスク

タスク	説明	機能
sessctrl	セッションコントローラ	StarOS の起動時に作成されます。システム内で一度にインスタンス化されるセッションコントローラは1つのみです。
		セッションサブシステムのプライマリコンタクトポイントとして機能します。システム内で実行されている他のサブシステムを認識しているため、セッションコントローラは、サブシステムを構成する他のコンポーネント、またはタスクのプロキシとして機能します。
		セッション処理サブシステムのサブマネージャの作業を開始、設定、および調整します。
		Resource Manager と連携して、既存のすべてのセッションマネージャがキャパシティを超えたときに新しいセッションマネージャを起動します。
		VPN マネージャからコンテキスト情報を受信します。
		他のセッション処理サブシステムサブマネージャに IP インターフェイスアドレス情報を配布します。
		Enhanced Charging Service (ECS)、コンテンツフィルタリング、および URL ブラックリストの各サービスを管理します。

タスク	説明	機能
sessmgr	セッションマネージャ	セッションコントローラによって作成されます。
		複数のセッションタイプをサポートするサブスクリバ処理システムを提供します。
		複数のセッションマネージャは、1つのCPU上で実行することも、システムに存在するすべてのCPUに分散することもできます。
		1つのセッションマネージャは、複数のA11マネージャか、または複数のコンテキストからセッションにサービスを提供できます。
		A10/A11、GRE、R3、R4、R6、GTPU/GTPC、PPP、およびモバイルIPのプロトコルを処理します。
		Enhanced Charging サービス、コンテンツフィルタリング、およびURLブラックリストの各サービスを管理します。
		セッションマネージャはAAAマネージャとペアになっています。
allmgr	A11 マネージャ	制限：AAAMGRが頻繁にクラッシュするとAAAmgrインスタンスIDが変更されるため、sessmgrが512を超えて増加する可能性があります。  たとえば、sessmgr インスタンス 512 以降に接続するコールの場合、符号化は10ビット（GUTIベースのMME接続時）、復号は9ビット（SGSN-CONTEXT-REQUEST時）であるため、context-request が誤った sessmgr にランディングしたり、ルックアップが失敗する原因となります。
		PDSN サービスが設定されているコンテキストごとに、セッションコントローラによって作成されます。
		PCFからのR-Pセッションを受信し、それらをロードバランシングのために別のセッションマネージャのタスクに配布します。
		システムリカバリを支援するため、現在のセッションマネージャのタスクのリストを維持します。
		A11 マネージャタスクは、シグナリングの逆多重化タスク（SDT）とも呼ばれます。  セッションリカバリ（SR）が有効になっている場合、このデマルチプレクサマネージャは通常、最初のアクティブなパケット処理カードのいずれかのCPU上に確立されます。

タスク	説明	機能
aaamgr	認証、許可、およびアカウントティング (AAA) マネージャ	セッションマネージャとペアになっています。
		システム内のサブスクリバと管理ユーザーに対して、すべての AAA プロトコルの操作と機能を実行します。
		AAA サーバーに対する AAA クライアントとして機能します。
		課金ゲートウェイ機能 (CGF) を使用して GTP プライム (GTP) メッセージングを管理します。
		複数の AAA マネージャを 1 つの CPU 上で実行することも、システムに存在するすべての CPU に分散することもできます。
		CLI の AAA 操作は、アクティブな管理カード上で実行されている AAA マネージャを介して行われます。
aaaproxy	認証、許可、およびアカウントティング (AAA) プロキシマネージャ	グローバル コンフィギュレーション モードの <b>gtp single source</b> コマンドが設定されるたびに起動します。  GTPP のシングルソーシングが有効になっている場合、aaaproxy は 1 つの UDP 送信元ポート番号を使用してアカウントティングサーバーへの要求を生成します。その一方で、各 AAA マネージャは、一意の UDP 送信元ポート番号を持つ独立した要求を生成する必要はありません。
		セッションリカバリが有効になっている場合は、デマルチプレクサカード上で実行します。セッションリカバリが有効になっていない場合、グローバルコンフィギュレーションモードの <b>demux card</b> コマンドを実行すると、指定したデマルチプレクサカードで aaaproxy が起動します。
		CDR を、その VRAM ディスク内のファイルに書き込みます。その後、キューに入れられた CDR は HDD と定期的に同期され、転送されます。

タスク	説明	機能
acsctrl	アクティブ課金システム (ACS) コントローラ	アクティブ課金サービスはグローバルレベルで定義され、任意の VPN コンテキストから CSS コマンドを使用して利用できます。グローバル コンフィギュレーション モードの <b>active-charging service</b> CLI コマンドを使用して有効にします。
		ACS コントローラはプライマリパケット処理カード上で動作し、ACS サービスを管理します。
		ACS の設定情報を読み取って、SCT に書き込みます。
		ACS コントローラは、ACS マネージャのリカバリプロセスをモニターし、冗長性が有効になっている場合はクリーンアップを実行します。
acsmgr	アクティブ課金システム (ACS) コントローラ	特定数のフローに対して IP セッション処理を実行するために、ACS コントローラによって作成されます。
		セッションマネージャを介してデータを送受信します。
		アクティブまたはスタンバイの acsmgr のタスクは、セッションリカバリ (SR) が有効になっている場合に作成されます。
cdrmod	課金詳細レコードモジュール	システム内のさまざまな ACSMGR インスタンスから EDR/UDR レコードを受信します。
		設定したファイル命名規則を使用して、受信した EDR/UDR レコードをファイルに書き込みます。
dgmbmgr	Diameter Gmb インターフェイス アプリケーション マネージャ	GGSN のマルチメディアブロードキャスト/マルチキャストサービス (MBMS) 機能をサポートします。これは、MBMS ポリシー CLI が GGSN サービス コンフィギュレーション モードで設定されている場合にインスタンス化されます。dgmbmgr
		MBMS UE とベアラークontextを維持します。
		MBMS ベアラークontextの BMSC サーバーへの Diameter 接続を介して Gmb インターフェイスを処理します。dgmbmgr は、すべての sessmgrs に対して MBMS のセッション状態をポーリングし、MBMS UE と MBMS ベアラークontextの情報を再作成することで回復します。

タスク	説明	機能
diamproxy	Diameter プロキシ	diactrl (vpnctrl の一部として実行) によって作成されます。また、生成される diamproxy タスクの数は、「multiple」または「single」プロキシを使用するための設定に基づいています。1つのプロキシが設定されている場合、シャース全体に対して1つの diamproxy タスクだけが生成され、デマルチプレクサパケット処理カード上で動作します。複数のプロキシが設定されている場合、1つの diamproxy タスクがパケット処理カードごとに実行されます。
		システムに設定されているすべてのピアへの Diameter ベースの接続を維持します。
		接続ステータスの変更に関するアプリケーションを通知します。
		アプリケーションから Diameter サーバーへのメッセージに対するパススルーとして機能します。
		転送エージェントとしてのみ機能します（キューは保持しません）。
		1つの Diameter プロキシを使用して、複数の Diameter アプリケーションにサービスを提供します。
egtpemgr	拡張 GPRS トンネリング プロトコル出力マネージャ	インターフェイスタイプの sgw-egress または MME の egtp-service が設定されているコンテキストごとにセッションコントローラが作成されます。
		SGW、PGW からの特定の EGTP メッセージを処理します。
		現在の EGTP セッションのリストを維持します。
		セッションリカバリに役立つ現在のセッションマネージャのタスクのリストを維持します。
		GTP エコーメッセージを処理します。
		セッションリカバリ（SR）が有効になっている場合、このデマルチプレクサマネージャは通常、最初のアクティブなパケット処理カードのいずれかの CPU 上に確立されます。

タスク	説明	機能
egtpimgr	拡張 GPRS トンネリング プロトコル入力マネー ジャ	インターフェイスタイプの <code>sgw-ingress</code> または <code>pgw-ingress</code> の <code>egtp-service</code> が設定されているコンテキストごとにセッションコントローラが作成されます。
		MME/S4 SGSN/SGW からの EGTP セッションを受信し、それらをロードバランシングのために別のセッションマネージャのタスクに配布します。
		現在の EGTP セッションのリストを維持します。
		セッションリカバリに役立つ現在のセッションマネージャのタスクのリストを維持します。
		GTP エコーメッセージを処理します。
		セッションリカバリ (SR) が有効になっている場合、このデマルチプレクサマネージャは通常、最初のアクティブなパケット処理カードのいずれかの CPU 上に確立されます。
gtpcmgr	GPRS トンネリングプロ トコル制御 (GTP-C) メッセージマネージャ	GGSN サービスが設定されているコンテキストごとに、セッションコントローラによって作成されます。
		SGSN からの GTP セッションを受信し、それらをロードバランシングのために別のセッションマネージャのタスクに配布します。
		システムリカバリを支援するため、現在のセッションマネージャのタスクのリストを維持します。
		GTPC メッセージの有効性を確認します。
		現在の GTPC セッションのリストを維持します。
		SGSN との間の GTPC エコーメッセージを処理します。

タスク	説明	機能
gtpumgr	GPRS トンネリングプロトコルユーザー (GTP-U) マネージャ	GTPU サービスが設定されているコンテキストごとに、セッションコントローラによって作成されます。GTPUv0 と GTPUv1 の両方でサポートされています。
		コンテキスト内で使用可能な GTPU サービスのリストを維持し、それらに対してロードバランシング (Error-Indのみ) を実行します。
		GTPU エコー処理をサポートします。
		GTPU エコーの応答がない場合のパス障害検出を実行します。
		Error-Indを受信し、特定のセッションマネージャに対して逆多重化します。
		デフォルトの GTPU リスナーとして機能します。GTPUMGR は、無効な TEID を持つ GTPU パケットを処理します。
		セッションリカバリ (SR) が有効になっている場合、このデマルチプレクサマネージャは通常、最初のアクティブなパケット処理カードのいずれかの CPU 上に確立されます。
hamgr	ホームエージェント (HA) マネージャ	HA サービスが設定されているコンテキストごとに、セッションコントローラによって作成されます。
		外部エージェント (FA) からモバイル IP セッションを受信し、それらを別のセッションマネージャのタスクに配布します。
		システムリカバリを支援するため、現在のセッションマネージャのタスクのリストを維持します。
		DemuxMgr としての機能し、すべての PMIP シグナリングパケットを処理します。
		MIPv6/MIPv4 HA の Demuxmgr として機能します。
		セッションリカバリ (SR) が有効になっている場合、このデマルチプレクサマネージャは通常、最初のアクティブなパケット処理カードのいずれかの CPU 上に確立されます。

タスク	説明	機能
hnbdemux	ホームノード b (HNB) デマルチプレクサマネージャ	HNB-GW サービスの作成手順の一環として開始します。hnbdemux はシャードに 1 つしかありません。
		着信した Iuh た接続をシステム内の HNB マネージャに配布します。
		システム内のアクティブなすべての HNB-GW サービスを認識しています。
		セッションリカバリ (SR) が有効になっている場合、このデマルチプレクサマネージャは通常、最初のアクティブなパケット処理カードのいずれかの CPU 上に確立されます。
hnbmgr	ホームノード b (HNB) マネージャ	HNB-GW サービス設定が検出されると開始します。ロード共有の場合は、このタスクの複数のインスタンスが存在する場合があります。すべての HNB マネージャには、アクティブなすべての HNB-GW サービスが設定されており、設定と機能が同一である必要があります。
		SCTP プロトコルスタックを実行します。
		SCTP の関連付けを処理します。
		ホームノード b データベースを維持します。
		SCTP プロトコルの Iuh インターフェイスにノード機能を提供します。
		セッションリカバリ (SR) が有効になっている場合、このマネージャは通常、最初のアクティブなパケット処理カードのいずれかの CPU 上に確立されます。



タスク	説明	機能
imsimgr	MME 用の International Mobile Subscriber Identity マネージャ	MME サービス設定が検出されると開始されます。このタスクのインスタンスは 1 つのみです。
		新しいサブスクリイバセッションに使用する SessMgr を選択します。
		IMSI による接続、GUTI による接続などのイベントについて、MME 関連のデマルチプレクサ統計情報を維持および報告します。
		システム内の次のタスクと連携します。 - セッションコントローラ - MME マネージャ - セッションマネージャ
		セッションリカバリ (SR) が有効になっている場合、このデマルチプレクサマネージャは通常、最初のアクティブなパケット処理カードのいずれかの CPU 上に確立されます。
imsimgr	SGSN 用の International Mobile Subscriber Identity マネージャ	セッションコントローラによって開始されます。
		IMSI/P-TMSI に基づくコールセッションで、linkmgr タスクまたは sgtpcmgr タスクによって実行されない場合は SessMgr を選択します。
		SessMgrs 間でロードバランシングを行い、サブスクリイバが割り当てられるものを選択します。
		システム上のすべてのサブスクリイバのレコードを維持します。
		IMSI/P-TMSI と SessMgrs 間のマッピングを維持します。
		セッションリカバリ (SR) が有効になっている場合、このデマルチプレクサマネージャは通常、最初のアクティブなデマルチプレクサパケット処理カードのいずれかの CPU 上に確立されます。
ipsgmgr	IP サービス ゲートウェイ マネージャ	セッションコントローラによって作成されます。
		サーバーモードでは、RADIUS サーバーとして機能し、プロキシ機能をサポートします。
		Snoop モードでは、RADIUS アカウンティングメッセージのスヌーピングをサポートします。
		異なる SessMgr 間で要求をロードバランシングします。
		セッションをアクティブまたは非アクティブにします。

タスク	説明	機能
l2tpdemux	L2TP デマルチプレクサタスク	LNS サービスが作成されたときにセッションコントローラによって作成されます。システム全体に対して1つのL2TPDemux タスクのみが呼び出されます。
		逆多重化を行い、新しい着信トンネルの作成要求をL2TPMgr に転送します。
		すべてのL2TPMgr で現在アクティブなトンネルに関する情報を維持します。
		L2TPMgr 間での要求をロードバランシングします。
		セッションリカバリ (SR) が有効になっている場合、このデマルチプレクサマネージャは通常、最初のアクティブなパケット処理カードのいずれかのCPU上に確立されます。
l2tpmgr	レイヤ2 トンネリングプロトコルマネージャ	LAC サービスまたはLNS サービスが設定されているコンテキストごとに、セッションコントローラによって作成されます。追加のマネージャは、ロードに応じ、必要に応じて作成されます。
		L2TP 処理のすべての側面を担います。
		すべてのL2TPセッションとトンネルのプロトコルステートマシンを維持します。
		必要に応じて、新しいL2TP トンネルのIPSec暗号化をトリガーします。
		セッションマネージャと連携して、トンネルを正常にダウンさせます。
		セッションリカバリ (SR) が有効になっている場合、このデマルチプレクサマネージャは通常、最初のアクティブなパケット処理カードのいずれかのCPU上に確立されます。
linkmgr	SS7 リンクマネージャ	最初のSS7RD (ルーティングドメイン) がアクティブになったときに、セッションコントローラによって作成されます。
		冗長性を得るためとスケーリングを行うためにマルチインスタンス化されます。
		プラットフォームへのSS7接続とGb接続を可能にします。
		SS7 (luを含む) とGbインターフェイス上のサブスクライバシグナリングごとにSessMgrへルーティングします。

タスク	説明	機能
magmgr	モバイル アクセス ゲートウェイ (MAG) マネージャ	最初の MAG サービスがコンテキストで作成されたときに、セッションコントローラによって作成されます。
		PMIP 制御メッセージ (PBU/PBA) を送受信します。
		MIPv6 PBA パケットを受信するために NPU フローを追加します。このフローは、HAMgr で使用されているフローと同じです。
		モバイルノードのバインディングを追跡し続けるために使用するバインディング更新リストを維持します。
		エラー状態時に、セッションマネージャから受信したトリガーに基づく PBU を発信します。
		PBA を受信し、セッションマネージャに転送します。
		デバッグファシリティ (「magmgr」と「mobile-ipv6」) をサポートします。
mmgr	SGSN マスターマネージャ	SS7RD や SCCP-NW などのプロビジョニング時に作成されます。セッションコントローラは、各分散プロトコルレイヤの詳細な説明、そのリソースセット、およびサービスユーザープロトコル層とサービスプロバイダープロトコル層のリストを含む、初期システム設定を提供します。
		1つのインスタンスとして実行されます。
		Nodal SS7、Iu、および Gb の機能を処理します。
		SS7 ルートステータスの集約のためのマスターリンクマネージャ機能を実装します。
		RNC ステータスと BSC ステータスの集約のためのマスターリンクマネージャ機能を実装します。
		セッションリカバリ (SR) が有効になっている場合、このデマルチプレクサマネージャは通常、最初のアクティブなデマルチプレクサパケット処理カードのいずれかの CPU 上に確立されます。

タスク	説明	機能
mmedemux	モビリティ マネージメント エンティティ デマルチプレクサ マネージャ	MME サービスの作成手順の一環として開始されます。シャーシには 1 つの mmedemux のみが存在します。
		着信 S1-MME SCTP 接続をシステム内の mmemgr タスクに配布します。
		システム内のアクティブなすべての MME サービスを認識しています。
		セッションリカバリ (SR) が有効になっている場合、このデマルチプレクサ マネージャは通常、最初のアクティブなパケット処理カードのいずれかの CPU 上に確立されます。
mmemgr	モビリティ マネージメント エンティティ マネージャ	MME サービス設定が検出されると開始されます。ロード共有の場合は、このタスクの複数のインスタンスが存在する場合があります。すべての mmemgrs マネージャには、アクティブなすべての MME サービスが設定されており、設定と機能が同一である必要があります。
		SCTP プロトコルスタックを実行します。
		SCTP の関連付けを処理します。
		TA リストを維持します。
		eNodeB データベースを管理します。
		S1-MME プロトコルのノード機能を提供します。
		セッションリカバリ (SR) が有効になっている場合、このデマルチプレクサ マネージャは通常、最初のアクティブなパケット処理カードのいずれかの CPU 上に確立されます。

タスク	説明	機能
pccdemux	ポリシーおよび課金制御 Bindmux マネージャ	PCC サービスの作成手順の一環として開始されます。シャードには、BindMux MGR のインスタンスが 1 つのみ存在します。
		セッションバインディング機能とともに、使用可能な pccmgrs 全体でセッションの多重化を処理します。
		pccmgrs での負荷をモニターします。
		着信 IP-CAN 接続をシステム内の pccmgrs 間に分散します。
		セッションのバインディングを実行します。IP-CAN/ゲートウェイセッションを AF セッションとバインドします。
		さまざまなインターフェイス間の IMSI のすべてのメッセージングが、選択した pccmgr に向けられるようにします。
		システム内のアクティブなすべての PCC サービスを認識しています。
		セッションリカバリ (SR) が有効になっている場合、このデマルチプレクサマネージャは通常、最初のアクティブなパケット処理カードのいずれかの CPU 上に確立されます。
pccmgr	ポリシーおよび課金制御 Bindmux マネージャ	pccmgr はセッション マネージャ インスタンスの一部です。
		すべての PCRF サービスセッションを処理します。
		個々のサブスクリバセッションに関連付けられているさまざまなイベント処理時に PCC コアと連動します。
		ビジネスロジックの適用時にサブスクリバ情報を維持します。
		サブスクリバごとに calline および対応する APN セッションを作成します。

タスク	説明	機能
sgtpcmgr	SGSN GPRS トンネリングプロトコル制御メッセージマネージャ	SGSN サービスが設定されている VPN コンテキストごとに、セッションコントローラによって作成されます。
		SGSN サービスのピア GGSN と SGSN からの Gn/Gp および GTP-U インターフェイスを終了します。
		IuPS サービスの RNC からの GTP-U インターフェイスを終端します。
		GTP-C と GTP-U の標準ポートを制御します。
		これらのポートでピアから受信した GTP トラフィックを処理して配布します。
		Gn/Gp インターフェイスに関連付けられているノードレベルのすべての手順を実行します。
		セッションリカバリ (SR) が有効になっている場合、このデマルチプレクサマネージャは通常、最初のアクティブなデマルチプレクサパケット処理カードのいずれかの CPU 上に確立されます。
srb	標準ルーティングデータベース	Enhanced Charging サービスでコンテンツフィルタリングが有効になっている場合に、セッションコントローラによって 8 つの SRB が作成されます。これらの 8 つのタスクを開始するには、少なくとも 2 つのパケット処理カードが必要です。
		セッションコントローラからスタティックデータベースを受信します。各 srb タスクは 2 つのデータベースボリュームをロードします (1 つはプライマリ、もう 1 つはセカンダリ)。srb タスクでは、スタティック DB も保存されません。
		DB ボリュームと、それに保存されている CSI (カテゴリセットインデックス) に基づいて URL が評価および分類されます。
		ピアに障害が発生した場合に、ピアのロードを実行します。srb タスクとそのピアの両方に障害が発生した場合、セッションコントローラはロードを実行します。

## プラットフォームプロセス

表 36: プラットフォームプロセスのタスク

タスク	説明	機能
afctrl	ASR 5500 Fabric Controller	システムファブリックの全体的な管理を担当します。ランデブーの宛先のプールを管理し、障害後の afmgr proclts によるファブリックのリカバリを調整します。単一の afctrl インスタンスは、アクティブな MIO/UMIO でのみ実行されます。
afmgr	ASR 5500 Fabric Manager	特定のカード上のファブリックリソースの管理を担当します。1つ以上のファブリック アクセス プロセッサ (FAP) またはファブリック要素 (FE) を担当するすべての CPU には、1つの afmgr があります。afmgr はリカバリをサポートしますが、移行はサポートしていません。
afio	ASR 5500 Fabric I/O Driver	ファブリックチップセットの直接設定を行います。afio は、ローカル afmgr と他のローカル afio インスタンスとのメッセージ以外のプロセス間通信 (IPC) をサポートしています。
connproxy	TCP/SCTP 接続プロキシ	カード上のアプリケーションは、カード上の各アプリケーションに対して新しい接続を開くのではなく、同じリモートエンドポイントに対して同じ TCP/SCTP 接続を共有できます。
cspectrl	カードスロットポートコントローラ	物理シャーシのコンポーネントを管理します。

タスク	説明	機能
cssctrl	Content Server Selection (CSS) コントローラ	コンテキスト内のサービスにバインド可能な CSS サーバーのリストを含むすべてのグローバル CSS プロパティを維持します。
		CSS は、モバイルサブスクリバによって提供されるか、モバイルサブスクリバに送信されるデータの「コンテンツ」に基づいてトラフィックを処理する方法を定義します。CSS には、ロードバランシング、NAT、HTTP リダイレクション、DNS リダイレクションなどの機能が含まれています。
		コンテンツサーバー (サービス) は、プラットフォームの外部に置くことも、プラットフォーム内に統合することもできます。外部 CSS サーバーは、コンテキスト コンフィギュレーション モードの <b>css server</b> コマンドを介して設定されます。
cssmgr	Content Server Selection (CSS) マネージャ	StarOS コンテキスト内の VPN マネージャによって生成されます。
		特定の VPN コンテキスト内の CSS サーバーへのキープアラートを管理します。
		サブスクリバの CSS 関連情報を取得します。
		CSS サーバーがダウンすると、cssmgr タスクは NPU を再プログラムしてサービスをバイパスしたり、サービス内の残りのサーバー間でデータを再配布したりします。
dcardctrl	ドーターカードコントローラ [ASR 5500 のみ]	システムの初期化中にドーターカードマネージャを生成し、システムが安定状態で実行中に、ドーターカードマネージャをモニターします。また、ドーターカードマネージャのタスクが失敗するたびに、ドーターカードマネージャを生成します。
dcardmgr	ドーターカードマネージャ [ASR 5500 のみ]	AH ベースおよび ESP ベースのセッションの IPSec セキュリティ アソシエーションの管理を担当します。
		特定の IPSec セキュリティ アソシエーションに関連付けられた暗号化アルゴリズムを実行するオンボードハードウェアアクセラレーションの暗号チップを備えたインターフェイス。



タスク	説明	機能
dhmgr	分散型ホストマネージャ	SITPARENT によって各 CPU で自動的に開始されます。
		sessmgr タスク間の Diameter エンドポイントなど、マルチインスタンス化されたタスクに代わって、ローカルで終端された TCP、SCTP、および UDP 接続の確立を調整します。
drvctrl	ドライバコントローラ	多くのシステムデバイスドライバへのアクセスを一元化します。また、温度と電圧のモニタリングも行います。
hdctrl	ハードドライブコントローラ	管理カードにまたがるドライブアレイを制御および管理します。
hwctrl	ハードウェアコントローラ	hwctrl タスクには、ハードウェアセンサーの読み取り、センサーのしきい値のモニタリング、およびファントレイのモニタリングのポーリンググループを管理する複数のタイマーがあります。
hwmgr	ハードウェアマネージャ	hwmgr タスクは、シャーシ内のすべてのカードで実行され、ローカルアクセス可能なハードウェアセンサーを読み取り、hwctrl に報告を返します。
inetd	InterNET Service Daemon	ほとんどのネットワークサービスの開始を担当するサブシステム。
		FTP、SFTP、Telnet などの接続クライアントからの要求をリッスンします。TCP パケットまたは UDP パケットが特定の宛先ポート番号で着信すると、inetd は適切なサーバープログラムを起動して接続を処理します。 注：FTP と Telnet はサポートされていません。
ipsecctrl	IPSec コントローラ	設定に関係なく、システムのスタートアップ時に SIT によって開始されます。
		設定に基づいて ipsecmgr タスクを開始し、タスクリカバリのリストを維持します。
		IPSec のユーザー設定を受信して維持します。
		設定された IPSec 暗号マップと ipsecmgrs への割り当てを管理します。
		IP アクセスリスト、IP プール、インターフェイスアドレス、インターフェイスの状態通知などの必要な IPSec 設定パラメータについて、vpnmgr タスクを使用したインターフェイス。
ipsecmgr	IPSec マネージャ	セッションコントローラによって作成され、セキュアな IKEv1、IKEv2、および IPSec データトンネルを、確立および管理します。

タスク	説明	機能
kvctrl	キー値コントローラ	管理カード上で実行される Central key value ストア (kvstore) 機能。その主な機能は、リカバリ機能および分散機能をサポートすることです。
lagmgr	リンク集約グループマネージャ [ASR 5500 のみ]	demux カードのプライマリ MIO (ASR 5500) の npuctrl によって起動し、CSP と npumgr の間のファシリティレベルで、npumgr から設定とステータス通知を受信して、グローバル LAG データベースを構築します。  設定された物理ポートを介して制御パケット (LACP および マーカー) をピアと交換し、リンクの集約に関する合意に到達します。
msgd	メッセージャデーモン	ネームサービスと、内部メッセージパッシングシステムの関連機能を実装します。
msgproxy	メッセージプロキシ	メッセージャプロキシプロセスは、単一のアプリケーション (クライアントと呼ばれる) から、スレッドごとに 1 つのインスタンスを持つファシリティ (ターゲットファシリティと呼ばれる) に送信するブロードキャストメッセージを処理します。  1 つの msgproxy タスクは、DPC (ASR 5500) 、および SF 仮想マシン (VPC-DI) 上の各 CPU コンプレックスで実行されます。  sessctrl などのクライアントプロセスからの着信ブロードキャストメッセージを処理して、sessmgr などの正しいターゲットファシリティに配信し、正しい応答を作成して正しいクライアントに送り返します。
nscontrol	ネームサービスコントローラ	メッセージャプロセスの一環として、メッセージャデーモンに制御メッセージを送信するタスクの信頼性の高いチャンネルを提供します。
ntpd	Network Time Protocol (NTP) デーモン	NTP を使用したタイムサーバーとの同期のシステム時刻を維持します。NTP コンフィギュレーション モードの <b>ntp server</b> CLI コマンドを使用して 1 つ以上の NTP サーバーが設定されている場合に有効になります。
rct	リカバリ制御タスク	システム全体のタスクとマネージャとファシリティをモニターし、障害が発生した場合にリカバリを実行します。
sct	共有設定タスク	インメモリデータベース内の設定情報およびその他の状態情報の冗長ストレージを実行します。
sft	スイッチファブリックタスク	スイッチファブリックとギガビットイーサネットコントロールプレーンをモニターします。

タスク	説明	機能
sshd	セキュアシェルデーモン	StarOS CLI へのセキュアログインをサポートします。コンテキストコンフィギュレーションモードの <b>server sshd</b> CLI コマンドを介して有効にします。
ucm	ユーティリティ設定マネージャ	DHCPD、DNS、FTPD、INETD、NTPD、PING、RLOGIN、SFTPD、SFTP-SERVER、SNMPD、SSH、SSHD、TELNET、TELNETD、TFTPD、TRACEROUTE 注：FTP と Telnet はサポートされていません。

## 管理プロセス

表 37: 管理プロセスのタスク

タスク	説明	機能
bulkstat	バルク統計情報マネージャ	バルク統計情報を定期的にポーリングして収集し、このデータを外部管理システムに転送します。
evlogd	イベントログデーモン	外部 syslog サーバーと内部イベントログへのインターフェイスを含むイベントロギング機能进行处理します。
orbs	ORBEM サービス [ASR 5500 のみ]	ORBS タスクは、ORB 要素マネージャ (ORBEM) と呼ばれています。  要素管理システム (EMS) は、セキュアな IOP を使用して要素管理機能をシステム上で実行するように ORBS に要求します。ORBS は、関連するコントローラのタスクと連携して機能を実行します。  実行による応答またはエラーが解釈され、EMF 応答に組み込まれて、EMS サーバーに渡されます。
orbns	ORBEM 通知サービス [ASR 5500 のみ]	イベントの発生を EMS サーバーに通知します。  このような EMS サーバーを登録し、関連するイベントタイプをサブスクライブします。  イベントが発生すると、関連するコントローラタスクが orbs (ORBEM) に通知し、次にサブスクライブしている EMS サーバーに通知します。

タスク	説明	機能
sesstrc	セッショントレース収集 タスク	標準ベースのセッショントレース機能を実装します。
		CLI とシグナリングベースの両方サブスクライバトレースを管理します。トレースするメッセージを収集し、必要に応じてトレースファイルを生成します。必要に応じて、トレースファイルをトレース収集エンティティにアップロードします。
snmp	簡易ネットワーク管理プロトコル	設定されている場合はインバウンド SNMP 動作を処理し、有効になっている場合は SNMP 通知（トラップ）を送信します。
threshold	しきい値サーバー	設定されている場合、しきい値超過アラートのモニタリングを処理します。必要な統計情報/変数をポーリングし、状態を維持し、しきい値超過のログメッセージ/SNMP 通知を生成します。



## 付録 C

# NETCONF および ConfD

この章では、NETCONF および ConfD manager と呼ばれる StarOS プロセスについて説明します。

ここで説明する内容は、次のとおりです。

- [機能の概要と変更履歴 \(469 ページ\)](#)
- [概要 \(471 ページ\)](#)
- [ConfD の設定 \(472 ページ\)](#)
- [設定の確認 \(478 ページ\)](#)
- [YANG モデル \(485 ページ\)](#)
- [Show Support Details \(SSD\) \(486 ページ\)](#)
- [ConfD の例 \(486 ページ\)](#)
- [ConfD のアップグレードサポート \(492 ページ\)](#)
- [ECS コマンド用の CLI ベースの YANG モデル \(492 ページ\)](#)

## 機能の概要と変更履歴

### 要約データ

該当製品または機能エリア	すべて
該当プラットフォーム	ASR 5500 VPC-SI VPC-DI
機能のデフォルト	無効：設定が必要
このリリースでの関連する変更点	N/A

関連資料	<ul style="list-style-type: none"> <li>• <i>ASR 5500 System Administration Guide</i></li> <li>• <i>Command Line Interface Reference</i></li> <li>• VPC-DI システム管理ガイド</li> <li>• VPC-SI System アドミニストレーション ガイド</li> </ul>
------	---

### マニュアルの変更履歴



(注) リリース 21.2 および N5.5 よりも前に導入された機能の改訂履歴の詳細は示していません。

改訂の詳細	リリース
<p>ノード選択とロードバランシング (NSLB) の主要業績評価指標 (KPI) のキャプチャに対するサポートが追加されました。</p> <p>この機能を有効にし、これらの KPI を収集するために使用する間隔を設定するために NETCONF プロトコル コンフィギュレーション モードの <b>kpi</b> コマンドが追加されました。</p> <p>詳細については、<a href="#">kpi (474 ページ)</a> および <a href="#">show confdmgr コマンド (479 ページ)</a> の項を参照してください。</p>	21.6
<p>SNMP MIB アラートとアラームは、NETCONF 通知を介して送信できるようになりました。</p> <p>この機能を有効にするために、NETCONF プロトコル コンフィギュレーションモードの <b>netconf</b> コマンドに <b>snmp</b> キーワードが追加されました。</p> <p><b>show confdmgr</b> コマンド出力が拡張されました。</p>	21.3
<p>ConfD で、REST インターフェイスを介して取得されたバルク統計の運用データを収集できるようになりました。</p> <p>新しい StarOS バルク統計情報とサーバー ConfD 設定の YANG モデルがサポートされています。StarOS CLI を介したすべての更新が自動的に ConfD データベースに同期されます。CLI ベースの YANG モデルは、StarOS ECS (Enhanced Charging System) コマンドにのみ適用されます。</p> <p>NETCONF プロトコル コンフィギュレーションモードに <b>bulkstats</b>、<b>netconf</b>、および <b>rest</b> コマンドが追加されました。<b>autosave-config</b> コマンドは廃止されました。</p> <p><b>show confdmgr</b> コマンドにキーワード <b>model bulkstats</b> と <b>model confd</b> が追加されました。</p> <p><b>show confdmgr</b> コマンド出力が拡張されました。</p>	21.2

改訂の詳細	リリース
最初の導入。	21.2 よりも前

## 概要

StarOS は、Cisco Network Service Orchestrator (NSO) とともに設定および運用データを転送するための YANG データモデルをサポートするノースバウンド NETCONF インターフェイスを提供します。また、NSO 管理コンソールと通信するための ConfD マネージャ (confdmgr) も組み込まれています。

NETCONF (Network Configuration Protocol) は、IETF (RFC 6241) によって開発および標準化されたネットワーク管理プロトコルです。ネットワークデバイスの設定をインストール、操作、削除するためのメカニズムを提供します。この操作は、単純なリモートプロシージャコール (RPC) レイヤの最上位で実現されます。NETCONF プロトコルは、設定データとプロトコルメッセージに XML ベースのデータ符号化を使用します。プロトコルメッセージは、セキュアなトランスポートプロトコル上で交換されます。

ConfD は、デバイスを管理するための一連のインターフェイスを提供する、オンデバイス管理フレームワークです。ConfD フレームワークは、データモデルからすべての管理インターフェイスを自動的にレンダリングします。ConfD は完全な NETCONF 仕様を実装し、XML でエンコードされたコンテンツを使用して SSH で実行します。

ConfD は、外部認証を介して認証/認可されたアクセスのみを許可するように設定されています。confdmgr は、新しいユーザーを認証/認可するときに、ConfD が起動するためのスタンドアロン CLI モジュールを提供します。ConfD は、StarOS 認証を介して認可されたアクセスのみを許可するように設定されています。認証時に、ユーザーには権限レベル (0~15) が与えられます。この権限レベルは、YANG モデルで定義されているように、StarOS *secure admin*、*admin*、*operator*、および *inspector* にマッピングされます。StarOS は、各 ConfD 認証要求の CLI 認証イベントやステータスメッセージをログに記録します。

サウスバウンド側では、ConfD は、ConfD 管理エージェントによって提供される一連の API を介して呼び出された StarOS プロセスと通信します。ConfD コンフィギュレーションデータベース (CDB) は、オブジェクトを保存するために ConfD によって使用されます。StarOS は、ConfD 提供の API を使用してデータベースにアクセスします。StarOS CLI を介したすべての更新が自動的に CDB に同期されます。

YANG は、NETCONF ネットワーク コンフィギュレーションプロトコル向けのデータモデリング言語です。これは、設定データとネットワーク要素の状態データの両方をモデル化するために使用できます。YANG は、ネットワーク要素によって生成されるイベント通知の形式を定義するために使用することもできます。また、データモデルは、NETCONF プロトコル (RFC 6020) を介してネットワーク要素で呼び出すことができるリモートプロシージャコールの署名を定義することができます。YANG ファイルは StarOS の一部としてコンパイルされ、既存の StarOS でサポートされている CLI コマンドを組み込みます。

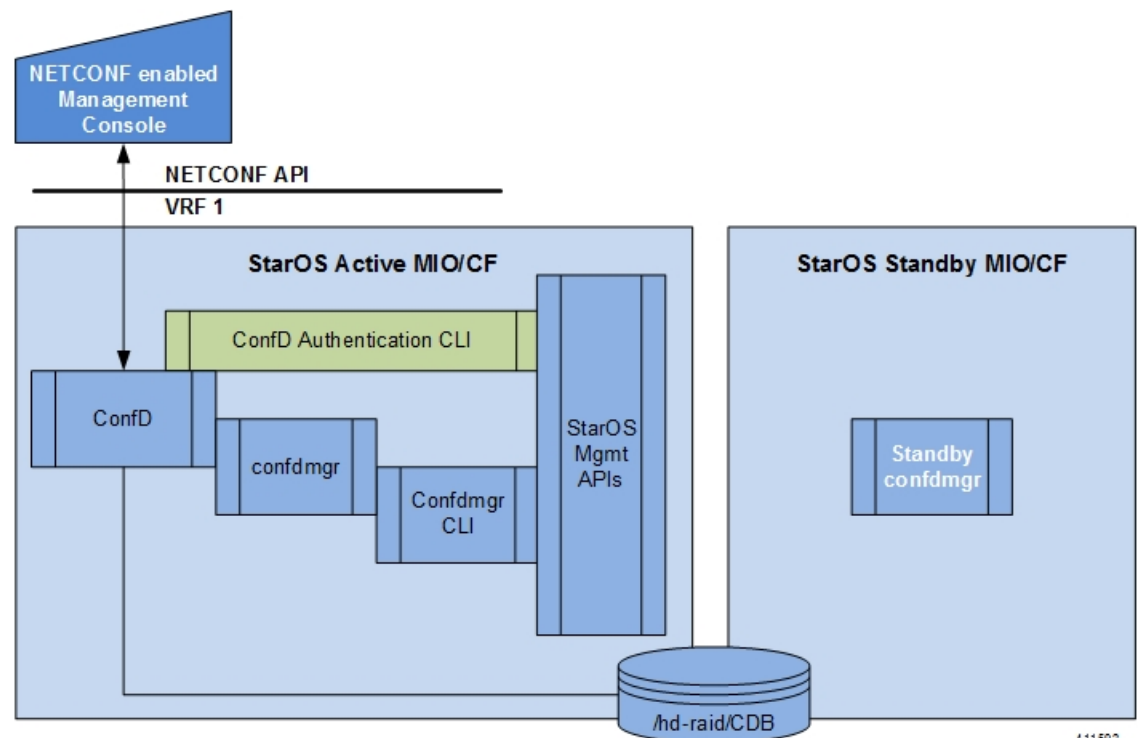
ConfD は、バルク統計情報の運用データを収集することもできます。有効にすると、StarOS は統計情報の収集時に confdmgr にスキーマ情報を送信します。収集されたバルク統計情報は、

REST (Representational State Transfer) インターフェイスを介して後で取得するために、ConfD CDB に保存されます。RESTCONF はトランスポートとして HTTP を使用して、YANG 仕様を RESTful インターフェイスにマッピングする方法を説明する IETF ドラフト

(draft-bierman-netconf-restconf-4) です。REST および RESTCONF は、有効な証明書とキーが設定されている場合にのみ内部で有効になります。クライアント認証が有効になっている場合は、CA 証明書も必要になることがあります。

NSO の詳細については、NSO のユーザーマニュアルを参照してください。

図 18: NETCONF システムフロー



411592

## ConfD の設定

StarOS で NETCONF プロトコルを有効にするには、**server confd** を有効にして、Netconf プロトコル コンフィギュレーション モードを開始する必要があります。NETCONF プロトコル コンフィギュレーションモードでは、オプションのコンフィギュレーションコマンドがサポートされています。

## SSH キーの要件

NETCONF-ConfD のサポートでは、ローカルコンテキストで V2-RSA SSH キーが設定されている必要があります。

SSH キーが使用できない場合、StarOS はエラーメッセージを生成します。



Failure: The ConfD (NETCONF) server requires an RSA key on the local context

**Show ssh key** コマンドを実行して、システム上の SSH キーが存在することを確認できます。

SSH キーが使用できない場合は、このガイドの「使用する前に」の章の「SSH オプションの設定」の項を参照してください。

## NETCONF プロトコル コンフィギュレーション モード

NETCONF プロトコルは、コンテキスト コンフィギュレーション モードの **erver conf** コマンドを介して有効にします。このコマンドは、ローカルコンテキストのみに制限されます。

```
[local]host_name# configure
[local]host_name(config)# context local
[local]host_name(config-ctx)# server confd
[local]host_name(config-confd)# ?

bulkstats          - Populate ConfD with bulkstats operational data
confd-user         - Configures the default login user with full administrator rights
                    for the ConfD server.
do                 - Spawns an exec mode command which displays
                    information to the administrator
end                - Exits configuration mode and returns to Exec Mode
exit               - Exits current configuration mode, returning to previous mode
kpi                - Key performance indicators gathering interval
netconf            - Configure the netconf interface
no                 - Enables/Disables the followed option
rest               - Configure the rest interface
```

次のキーワードはオプションです。

- **bulkstats**
- **confd-user**
- **kpi**
- **netconf**
- **rest**

NETCONF プロトコルを無効にするには、コンテキスト コンフィギュレーション モードの **no server confd** コマンドを実行します。

詳細については、『*Command Line Interface Reference*』の「*NETCONF Protocol Configuration Mode Commands*」の章を参照してください。

## bulkstats

この NETCONF プロトコル コンフィギュレーション モード コマンドを使用すると、REST インターフェイスを介したバルク統計情報の収集とレポートが可能になります。デフォルトでは、このコマンドはディセーブルです。

コマンドシンタックスは **bulkstats** です。

REST インターフェイスを介して後から取得するため、StarOS の統計情報の収集中にバルク統計情報が CDB にも保存されます。

バルク統計情報の運用データを使用した ConfD の入力を無効にするには、**no bulkstats** を使用します。

詳細については、『*Command Line Interface Reference*』の「*NETCONF Protocol Configuration Mode Commands*」の章を参照してください。

## confd-user

この NETCONF プロトコルのコンフィギュレーションモードのコマンドは、NETCONF を介してすべての CLI 操作のユーザー名を関連付けます。ユーザは、検証可能なログイン情報で認証されます。このユーザー名は、CLI ログイン目的でのみ使用されます。

コマンドシンタックスは、**confd-user <username>** です。ここで <username> は、1 ~ 144 文字の英数字の文字列です。




---

**重要** NETCONF セッションまたは RESTful セッションは、検証可能なログイン情報で引き続き確立する必要があります。

---

## kpi

この NETCONF プロトコルコンフィギュレーションモードでは、NSLB の重要業績評価指標 (KPI) の収集間隔を設定します。

コマンドシンタックスは **kpi seconds** であり、*seconds* は 0 (無効) または 10 ~ 120 の整数で、次の KPI を収集する時間間隔を秒単位で設定します。

- セッションによる CPU の使用率 (%)
- セッションによるメモリの使用率 (%)
- セッション以外による CPU の使用率 (%)
- セッション以外によるメモリ使用率 (%)
- セッションによる使用率 (%)

これらの統計情報は、システム全体で (すべてのカードで) キャプチャされます。デフォルトでは、この機能はディセーブルです。

詳細については、『*Command Line Interface Reference*』の「*NETCONF Protocol Configuration Mode Commands*」の章を参照してください。

## netconf notifications events

この NETCONF プロトコル コンフィギュレーション モード コマンドは、StarOS に記録された イベントを「StarOS」という名前のストリームで NETCONF 通知として送信できるようにします。Level では、通知が生成される最も低いイベントシビラティ（重大度）レベルを指定します。

コマンドシンタックスは **netconf notifications events level { critical | error | warning | unusual | info }** で、レベルは次のようになります。

- **critical** - Level 1 : ログファイルに重大なエラーが含まれていることを報告します。
- **error** - Level 2 : ログファイルにエラー通知が含まれていることを報告します。
- **warning** - Level 3 : ログファイルに警告メッセージが含まれていることを報告します。
- **unusual** - Level 4 : ログファイルに予期しないエラーが含まれていることを報告します。
- **info** - Level 5 : ログファイルに情報メッセージが含まれていることを報告します。

NETCONF 通知を無効にするには **no netconf notifications events** を使用します。



---

**重要** カテゴリ「critical-info」（シビラティ（重大度）を問わない）のイベントも通知に変換されません。

---

## netconf notifications snmp

この NETCONF プロトコル コンフィギュレーション モード コマンドは、SNMP アラートとアラームに「StarOS\_SNMP」という名前のストリームで NETCONF 通知として送信できるようにします。

コマンドシンタックスは **netconf notifications snmp** です。

NETCONF 通知を無効にするには **no netconf notifications snmp** を使用します。

## netconf port

この NETCONF プロトコル コンフィギュレーション モード コマンドは、NETCONF インターフェイスのポート番号を設定します。**server confd** が有効になっている場合、デフォルトポートは自動的に 830 に設定されます。

コマンドシンタックスは **netconf port port\_number** で、*port\_number* は 1 ~ 65,535 の整数である必要があります。

ポート番号を 830 にリセットするには **no netconf port** を使用します。



**重要** NETCONF インターフェイスのポート値を変更すると、ConfD の再起動が予定され、NETCONF および REST（有効になっている場合）インターフェイスを介した接続の一時的な切断が発生します。

## rest auth-policy

この NETCONF プロトコル コンフィギュレーション モード コマンドは、サーバーがクライアント証明書に対して実行する検証のレベルを制御します。CA（認証局）証明書は、グローバル コンフィギュレーション モードで既存の **ca-certificate** コマンドを使用して設定できます。

コマンドシンタックスは **rest auth-policy { none | peer | peer-fail }** で、レベルは次のようになります。

- **none** : 認証は実行されません。
- **peer** : クライアントが証明書を提供しない場合、またはクライアントが証明書を提供し、それが有効である場合は、接続が許可されます。クライアントが無効な証明書を提供した場合、接続は中断されます。



**重要** **peer** が選択されている場合は、CA 証明書が推奨されます。そうでない場合、有効な証明書を提供するクライアントは認証できず、接続は失敗します。

- **peer-fail** : サーバーはクライアントがクライアント証明書を提供するように要求し、証明書が正常に検証されない場合は接続に失敗します。



**重要** **peer-fail** が選択されている場合は、1 つ以上の CA 証明書がデバイス上に存在する必要があります。そうでない場合は REST インターフェイスは有効になりません。

**no rest auth-policy** を使用して認証ポリシーを **none** に設定します。認証は実行されません。



**重要** REST インターフェイスの認証ポリシーを変更すると、ConfD の再起動が予定され、NETCONF および REST（まだ有効になっている場合）インターフェイスを介した接続の一時的な切断が発生する可能性があります。

REST が有効になっている間に ConfD が使用しているグローバル証明書を変更すると、ConfD の再起動も発生させます。

## rest certificate

この NETCONF プロトコル コンフィギュレーション モード コマンドは、REST インターフェイスの証明書と秘密キーを設定します。

コマンドシンタックスは **rest certificate** *certificate\_name* で、*certificate\_name* は 1 ～ 128 文字の英数字の文字列です。



---

**重要** 指定した証明書がデバイス上に存在している必要があります。証明書および関連付けられた秘密キーは、既存の **certificate** コマンドをグローバル コンフィギュレーション モードで使用して設定できます。

設定した証明書とキーを削除するには **no rest certificate** を使用します。有効な証明書とキーがないと REST は動作しません。



---

**重要** REST インターフェイスの証明書を変更すると、ConfD の再起動が予定され、NETCONF および REST (有効になっている場合) インターフェイスを介した接続の一時的な切断が発生することがあります。

REST が有効になっている間に ConfD が使用しているグローバル証明書を変更すると、ConfD の再起動も発生させます。

---

## rest hostname

この NETCONF プロトコル コンフィギュレーション モード コマンドで Web サーバーが対応するホスト名を指定します。設定されている場合、Web サーバーは、ホストフィールドが設定されたホスト名と一致する要求にのみ対応するよう求められます。

コマンドシンタックスは **rest hostname** *host\_name* であり、*host\_name* は 1 ～ 63 文字の英数字の文字列です。

システム名を使用するには **no rest hostname** を使用します。ホスト名との一致は求められません。



---

**重要** REST インターフェイスのホスト名を変更すると、ConfD の再起動が予定され、NETCONF および REST (まだ有効になっている場合) インターフェイスを介した接続の一時的な切断が発生する可能性があります。

REST が有効になっている間に ConfD が使用しているグローバル証明書を変更すると、ConfD の再起動も発生させます。

---

## rest port

この NETCONF プロトコル コンフィギュレーション モード コマンドは、REST インターフェイスのポート番号を設定します。

コマンドシンタックスは **rest port port\_number** で、*port\_number* は 1 ~ 65,535 の整数である必要があります。

ポート番号をデフォルトの 443 にリセットするには **no rest port** を使用します。



**重要** REST インターフェイスのポート値を変更すると、ConfD の再起動が予定され、NETCONF および REST (まだ有効になっている場合) インターフェイスを介した接続の一時的な切断が発生する可能性があります。

REST が有効になっている間に ConfD が使用しているグローバル証明書を変更すると、ConfD の再起動も発生させます。

## 設定例

次のコマンドシーケンスは、NETCONF プロトコルをサポートする ConfD 設定を確立します。

**server confd** を有効にするには、タイプ v2-RSA SSH キーが必要です。

```
configure
  context local
    ssh key
  <encrypted key text>
  len 938 type v2-rsa
  server confd
    bulkstats
    confd-user NETCONF
    rest certificate rest-cert
  #exit
  subscriber default
  exit
  aaa group default
  #exit
  gtp group default
  #exit
#exit
end
```

注：

- **bulkstats**、**confd-user**、および **rest** はオプションです。**server confd** を設定すると、NETCONF サポートが有効になります。

## 設定の確認

NETCONF-ConfD 設定に関する情報を表示するには、2 つの Exec モードの **show** コマンドがあります。

## show confdmgr コマンド

このコマンドは、StarOS ConfD Manager (confdmgr) プロセスに関する情報を表示します。

このコマンドの構文は、次のとおりです。

```
show confdmgr [ confd { cdb | netconf | state } | model { bulkstats |
confd } | subscriptions ] [ | { grep grep_options | more } ]
```

注：

- **confd** キーワードは、次のオプションの指定されたキーワードに基づいて ConfD エンジンに関する情報を表示します。
  - **cdb** は ConfD CDB 情報を表示します。
  - **netconf** は NETCONF ステート情報を表示します。
  - **state** は現在の ConfD ステート情報を表示します。
- **model** キーワードは、次のオプションの指定されたキーワードに基づいて ConfD モデルに関する情報を表示します。
  - **bulkstats** : バルク統計情報の設定と運用データ
  - **confd** : サーバー ConfD 設定
- **サブスクリプション** キーワードを指定すると、ConfD CDB サブスクリプション情報が表示されます。

### show confdmgr

**show confdmgr** の出力例については、以下を参照してください。

```
[local]<host_name># show confdmgr

State Information
-----
State                Started
Subscriptions        5
Last successful id   1461-704882-705350
Last failed id      None
Username             Not configured
Bulkstats            Enabled
Kpi interval         30
Event notification level Disabled
SNMP notifications  Disabled
REST interface authentication none
REST interface certificate rest-cert
REST interface host name Not configured

Interface                Status          Port
-----
NETCONF                  Enabled         830
REST                     Enabled         443

Statistics
-----
```

```

Triggers                1
Replays                 0
Notifications           5
Notification failures   0
Trigger failures       0
Replay failures         0
NETCONF notification failures 0
Unexpected failures     0
[local]<host_name>#

```

この出力の統計部分には、次の情報が含まれています。

- **Triggers** : confdmgr が CDB の内容をダンプして confdmgr に戻すように ConfD に要求した結果、SCT への設定の同期が行われた回数（共有設定タスク）。
- **Replays** : トランザクションが再生された回数。再生は、起動時に confdmgr で最後に成功したトランザクション ID が ConfD と一致しない場合に開始されます。たとえば、これは設定トランザクションの通知を処理するとき confdmgr タスクが再起動された場合に発生する可能性があります。
- **Notifications** : ConfD が confdmgr に設定の更新を送信した回数。たとえば、これは confd\_cli を介した「commit」か、またはトリガーイベントの結果として発生することがあります。
- **Failed Notifications** : ConfD から受信した設定が正常に処理されなかった回数。
- **Trigger failures** : confdmgr への CDB ダンプが失敗した回数。
- **Replay failures** : トランザクションの再生試行が失敗した回数。
- **NETCONF notification failures** : NETCONF 通知の発行試行が失敗した回数。
- **Unexpected failures** : 予期しない状況が発生した回数。各ケースに対してエラーログが生成されます。

### show confdmgr confd cdb

show confdmgr confd cdb の出力例については、次を参照してください。

```

[local]<host_name># show confdmgr confd cdb
bulkstats server collection true
bulkstats server historical-collection false
bulkstats server gather-on-standby true
bulkstats server sample-interval 60
bulkstats server transfer-interval 1440
bulkstats server limit 7500
bulkstats server receiver-mode secondary-on-failure
bulkstats server file 1
!
bulkstats schemas file 1
schema-type system
  schema abc
  format      %host%
  active-only false
!
schema common
  format      %host%,%ipaddr%,%time%,%uptime%,%swbuild%,%localtz%
  active-only false
!
schema systemSch11

```



```

format
PPM,system,systemSch11,%epochtime%,%localdate%,%localtime%,%uptime%,%diameter-msg
-saans%,%diameter-msg-sarretry%,%diameter-msg-saatimeout%,%diameter-msg-saadropped%,%diameter-ms
g-uareq%,%diameter-msg-uaans%,%diameter-msg-uarretry%,%diameter-msg-uaatimeout%,%diameter-msg-ua
adropped%,%diameter-msg-lireq%,%diameter-msg-lians%,%diameter-msg-lirretry%,%diameter-msg-liatim
eout%,%diameter-msg-liadropped%,%diameter-msg-rtreq%,%diameter-msg-rtans%,%diameter-msg-rtrejec
t%,%diameter-msg-ppreq%,%diameter-msg-ppans%,%diameter-msg-ppreject%,%diameter-msg-dereq%

    active-only false
    !
    !
    !
confd bulkstats true
confd netconf port 830
confd rest port 443
confd rest auth-policy none
confd rest certificate rest-cert
nacm read-default permit
nacm groups group admin
!
nacm groups group inspector
!
nacm groups group operator
!
nacm groups group secure_admin
!
nacm rule-list secure_admin
!
group [ secure_admin ]
rule any-access
    action permit
!
rule secure_admin_server_confid
module-name      cisco-staros-cli-config
path             /context/server/confd
access-operations create,read,update
action           permit
|
|
V
nacm rule-list inspector
group [ inspector ]
rule any-access
    access-operations read
    action           permit
!
!
[local]<host_name>#

```

### show confdmgr confd netconf

**show confdmgr confd netconf** の出力例については、以下を参照してください。

```

[local]<host_name># show confdmgr confd netconf
netconf-state capabilities capability urn:ietf:params:netconf:base:1.0
netconf-state capabilities capability urn:ietf:params:netconf:base:1.1
netconf-state capabilities capability
urn:ietf:params:netconf:capability:writable-running:1.0
netconf-state capabilities capability urn:ietf:params:netconf:capability:candidate:1.0
|
|
V
netconf-state statistics netconf-start-time 2016-03-30T17:09:49-04:00
netconf-state statistics in-bad-hellos 0

```

```

netconf-state statistics in-sessions 0
netconf-state statistics dropped-sessions 0
netconf-state statistics in-rpcs 0
|
|
V
netconf-state datastores datastore candidate
netconf-state schemas schema cisco-staros-bulkstats 2016-12-14 yang
namespace http://www.cisco.com/staros-bulkstats
location [ NETCONF ]
netconf-state schemas schema cisco-staros-bulkstats-config 2016-12-14 yang
namespace http://www.cisco.com/staros-config
location [ NETCONF ]
|
|
V
NAME                                CREATOR   CREATED                                CONTEXT
-----
/rollback0                          system    2017-01-17T13:40:53-00:00             system
/rollback1                          system    2017-01-17T13:40:52-00:00             system
/rollback2                          system    2017-01-17T13:40:52-00:00             system
/rollback3                          system    2017-01-17T13:40:52-00:00             system
/rollback4                          system    2017-01-17T13:36:43-00:00             system
|
|
V
/cli-history/admin.hist
/cli-history/root.hist
/global.data

netconf-state streams stream NETCONF
description                          "default NETCONF event stream"
replay-support                       false
netconf-state streams stream StarOS
description                          "StarOS Notifications"
replay-support                       true
replay-log-creation-time 2017-02-10T16:00:59+00:00
[local]<host_name>#

```

### show confdmgr confd state

**show confdmgr confd state** の出力例については、以下を参照してください。

```

[local]<host_name># show confdmgr confd state
Monday June 24 10:58:49 EDT 2019
confd-state version 7.1
confd-state epoll false
confd-state daemon-status started
confd-state loaded-data-models data-model acs-config
revision                             2016-10-31
namespace                             http://www.cisco.com/usp/nfv/acs-config
prefix                                 acs-config
exported-to-all
confd-state loaded-data-models data-model cisco-staros-bulkstats
revision                             2016-12-14
namespace                             http://www.cisco.com/staros-bulkstats
prefix                                 staros_bulkstats
exported-to-all
confd-state loaded-data-models data-model cisco-staros-cli-config
revision                             2016-12-14
namespace                             http://www.cisco.com/staros-cli-config
prefix                                 staros_cli
exported-to-all
confd-state loaded-data-models data-model cisco-staros-config

```

```

revision          2016-12-14
namespace         http://www.cisco.com/staros-config
prefix           staros_config
exported-to-all
confd-state loaded-data-models data-model cisco-staros-exec
revision          2016-12-14
namespace         http://www.cisco.com/staros-exec
prefix           staros_exec
exported-to-all
confd-state loaded-data-models data-model cisco-staros-kpi
revision          2017-10-31
namespace         http://www.cisco.com/staros-kpi

```

### show confdmgr model bulkstats

**show confdmgr model bulkstats** の出力例については、次を参照してください。

```
[local]<host_name># show confdmgr model bulkstats
```

```

Model: Bulkstats
-----

Operational Data:
  Requests          277
  Records           831
  Failures           0

Configuration:
  CLI updates       0
  NETCONF updates   2
  Aborts            0
  Failures          0
local]<host_name>#

```

この出力の運用データ部分には、次の情報が含まれています。

- **Requests** : bulkstats から confdmgr への運用データメッセージ要求の数。
- **Records** : 処理された運用データスキーマレコードの数。
- **Failures** : バルク統計情報からのプッシュ要求の処理中に confdmgr 内で検出されたエラーの数。

この出力の設定部分には、次の情報が含まれています。

- **CLI updates** : CLI からのプッシュ設定要求の数と、SCT からの設定のロードの回数。
- **NETCONF updates** : バルク統計情報のサブスクリプション通知の数。
- **Aborts** : NETCONF を介した設定更新が中断された回数。
- **Failures** : confdmgr 内のバルク統計情報設定要求の処理で検出されたエラーの数。

### show confdmgr model confd

**show confdmgr model confd** の出力例については、次を参照してください。

```
[local]<host_name># show confdmgr model confd
```

```
Model: ConfD
```

```

-----
CLI updates      0
NETCONF updates  1
Aborts           0
Failures         0
local]<host_name>#

```

- **CLI updates** : CLI からのプッシュ設定要求の数と、SCT からの設定のロードの回数。
- **NETCONF updates** : ConfD 設定サブスクリプション通知の数。
- **Failures** : confdmgr 内の ConfD 設定要求の処理で検出されたエラーの数。
- **Aborts** : NETCONF を介した設定更新が中断された回数。

### show confdmgr subscriptions

**show confdmgr subscriptions** の出力例については、以下を参照してください。

```

[local]<host_name># show confdmgr subscriptions

Subscriptions:
Path                               Index  Namespace
-----
/active-charging-fig              6     http://www.cisco.com/staros-cli-con
/context-fig                       7     http://www.cisco.com/staros-cli-con
/bulkstats/server                 8     http://www.cisco.com/staros-config
/bulkstats/schemas               9     http://www.cisco.com/staros-config
/confd                             10    http://www.cisco.com/staros-config
[local<host_name>#

```

サブスクリプションとは、変更が発生したときに **confdmgr** に通知する必要がある Yang モデルで定義されている設定ポイントのことです。

## clear confdmgr confd cdb

この Exec モードコマンドは、ConfD が設定オブジェクトを保存するために使用する ConfD 設定データベース (CDB) の設定を消去します。StarOS は、ConfD が提供する API を使用してデータベースにアクセスします。



- (注) ConfD および NETCONF プロトコルのサポートを無効にするために、コンテキスト コンフィギュレーション モードの **no server confd** コマンドがローカルコンテキストで実行されている場合を除き、CDB を消去することはできません。

次に、CDB をクリアするためのコマンドシーケンスの例を示します。

```

[local]host_name# config
[local]host_name(config)# context local
[local]host_name(config-ctx)# no server confd
[local]host_name(config-ctx)# end
[local]host_name# clear confdmgr confd cdb
About to delete the ConfD configuration database

```

```
The running configuration is NOT affected.  
Are you sure? [Yes|No]: Y  
[local]host_name#
```



**注意** CDB のクリアは端末の動作です。CDB はコンテキスト コンフィギュレーション モードの **server confd** コマンドがローカルコンテキストで実行されているときに再度追加されて、ConfD と NETCONF プロトコルサポートを再度有効にします。

## clear confdmgr statistics

このコマンドは、**show confdmgr** コマンドの出力の [Statistics] セクションに掲載されているすべての内容がクリアされます。これには、次が含まれます。

- Triggers
- Replays
- Notifications
- Notification failures
- Trigger failures
- Replay failures
- NETCONF notification failures
- Unexpected failures

## YANG モデル

StarOS のインストールでは、次の YANG ファイルを使用できます。

- **cisco-staros-bulkstats-config.yang** : StarOS のネイティブバルク統計設定モデル。
- **cisco-staros-bulkstats-schema-types.yang** : **cisco-staros-bulkstats-config.yang** モデルの拡張機能。コードから直接取得したスキーマ名を列挙したリストが含まれています。
- **cisco-staros-bulkstats.yang** : RESTful インターフェイスを介して顧客がバルク統計を取得できるようにする運用データモデル。このモデルは、管理者クレデンシャルを持つユーザーのみが使用できます。
- **cisco-staros-confd-config.yang** : ネイティブのサーバー ConfD 設定モデル。
- **cisco-staros-config.yang** : 他のすべての **cisco-staros-\*** 設定モデルを含めるために使用されるコンテナ YANG ファイル (すべてのネイティブモデルが共通名前空間下のこの場所に含まれています)。

- **cisco-staros-exec.yang** : RESTful インターフェイスを介した CLI の実行動作を可能にするモデル。このモデルは、管理者クレデンシャルを持つユーザーのみが使用できます。入力を解析するために ConfD によってローカルに使用されます。
- **cisco-staros-notif.yang** : StarOS のイベントロギングの NETCONF 通知ストリームを可能にするモデル。デバッグレベルのイベントはサポートされていません。情報メッセージ以上のみがサポートされています。



**重要** これらの YANG ファイルが読み込まれて使用可能になる前に、ConfD サーバーを少なくとも 1 回起動する必要があります。

StarOS ネットワーク要素ドライバ (NED) を構築するには、YANG ファイルを Cisco NSO にプルする必要があります。

YANG ファイルをコピーするには、次のようなコマンドを入力します。

```
#copy /hd-raid/confd_dir/etc/confd/cisco-staros-confd-config.yang
sftp://<user>:<password>@<host>/sftp-directory/cisco-staros-confd-config.yang
```

## Show Support Details (SSD)

すべての **show confdmgr** コマンドの出力が SSD に追加されました。

## ConfD の例

### サーバー ConfD

次に、完全な TLS 認証と curl を使用してサーバー ConfD 設定を取得する例を示します。

#### サーバー ConfD の設定

次に、デフォルト以外の NETCONF と HTTPS ポートを使用して RESTful インターフェイスが有効になっているサーバー ConfD の設定例を示します。

```
[local]<host_name># show configuration confd
[local]<host_name># config
[local]<host_name>(config)# ca-certificate name ca-cert pem url /flash/ssl/rootCA.pem
[local]<host_name>(config)# certificate name rest-cert pem url /flash/ssl/host.crt
private-key pem url /flash/ssl/host.key
[local]<host_name>(config)# end
[local]<host_name># config
[local]<host_name>(config)# context local
[local]<host_name>(config-ctx)# server confd
[local]<host_name>(config-confd)# netconf port 123
[local]<host_name>(config-confd)# rest port 234
[local]<host_name>(config-confd)# rest certificate rest-cert
[local]<host_name>(config-confd)# rest auth-policy peer-fail
```

```
[local]<host_name>(config-confd)# end
[local]<host_name># show confdmgr

State Information
-----
State                               Started
Subscriptions                        5
Last successful id                   1488-211047-99241
Last failed id                       None
Username                             Not configured
Bulkstats                            Disabled
Event notification level             Disabled
SNMP notifications                   Disabled
REST interface authentication        peer-fail
REST interface certificate            rest-cert
REST interface host name             Not configured

Interface          Status          Port
-----
NETCONF            Enabled         123
REST               Enabled         234

Statistics
-----
Triggers           1
Replays            0
Notifications      8
Notification failures 0
Trigger failures   0
Replay failures    0
NETCONF notification failures 0
Unexpected failures 0
```

### netconf-console を使用したサーバー ConfD 設定の取得

次に、NETCONF を介してサーバー ConfD 設定を取得するための netconf-console の使用例を示します。

```
[user@server]$ ./netconf-console --host 1.2.3.4 -u admin --password pswd! --port 123
--get-config -x confd
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="1">
  <data>
    <confd xmlns="http://www.cisco.com/staros-config">
      <bulkstats>>false</bulkstats>
      <netconf>
        <port>123</port>
      </netconf>
      <rest>
        <port>234</port>
        <auth-policy>peer-fail</auth-policy>
        <certificate>rest-cert</certificate>
      </rest>
    </confd>
  </data>
</rpc-reply>
```

注：

- netconf-console は GitHub (<https://github.com/tail-f-systems/JNC/blob/master/examples/2-junos/netconf-console>) から自由に使用できます。

## Curl を使用したサーバー ConfD 設定の取得

次に、curl を使用して同じ **get-config** 操作を実行する例を示します。

```
[<user>@server] ]$ curl -u admin:pswd!
https://rtp-mitg-si06.cisco.com:234/api/running/confd?deep --cert
/users/<user>/ssl_cert/client_cert/client.crt --key
/users/<user>/ssl_cert/client_cert/client.key --cacert
/users/<user>/ssl_cert/root_cert/rootCA.pem

<confd xmlns="http://www.cisco.com/staros-config" xmlns:y="http://tail-f.com/ns/rest"
xmlns:staros_config="http://www.cisco.com/staros-config">
  <bulkstats>>false</bulkstats>
  <netconf>
    <port>123</port>
  </netconf>
  <rest>
    <port>234</port>
    <auth-policy>peer-fail</auth-policy>
    <certificate>rest-cert</certificate>
  </rest>
</confd>
```

## バルク統計情報

次に、バルク統計情報の運用データの例を示します。

### バルク統計情報の有効化

サーバー ConfD で **bulkstats** を有効にします。

```
[local]<host_name># config
[local]<host_name>(config)# context local
[local]<host_name>(config-ctx)# server confd
[local]<host_name>(config-confd)# bulkstats
[local]<host_name>(config-confd)# end
[local]<host_name># show confdmgr

State Information
-----
State                Started
Subscriptions        5
Last successful id   1488-216669-170664
Last failed id      None
Username             Not configured
Bulkstats            Enabled
Event notification level Disabled
SNMP notifications  Disabled
REST interface authentication peer-fail
REST interface certificate rest-cert
REST interface host name Not configured

Interface            Status          Port
-----
NETCONF              Enabled         123
REST                  Enabled         234
```

### バルク統計情報の設定

バルク統計情報の設定例については、次を参照してください。



```
[local]<host_name># show config bulkstats
config
  bulkstats collection
  bulkstats mode
    file 1
      schema common format %uptime%,%host%,%ipaddr%
    #exit
    file 2
      schema system format %uptime%,%host%,%ipaddr%
    #exit
  #exit
End
```

### バルク統計情報収集の強制

詳細情報を強制的に収集して、ConfDの運用データベースにプッシュする例については、次を参照してください。

```
[local]<host_name># bulkstats force gather
```

注：

- 統計情報は一般に、bulkstats に設定された収集間隔タイマーごとにプッシュされます。

### Curl を使用した統計情報の読み取り

サーバー ConfD RESTful インターフェイスを介して統計情報を読み取るための Curl の使用例については、次を参照してください。

```
[<user>@server] ]$ curl -u admin:pswd!
https://rtp-mitg-si06.cisco.com:234/api/operational/bulkstats-operational?deep --cert
/users/<user>/ssl_cert/client_cert/client.crt --key
/users/<user>/ssl_cert/client_cert/client.key --cacert
/users/<user>/ssl_cert/root_cert/rootCA.pem
```

```
<bulkstats-operational xmlns="http://www.cisco.com/staros-bulkstats"
xmlns:y="http://tail-f.com/ns/rest"
xmlns:staros_bulkstats="http://www.cisco.com/staros-bulkstats">
  <file>
    <number>1</number>
    <schemas>
      <schema>system</schema>
      <names>
        <name>common</name>
        <key_ids>
          <key_id>none</key_id>
          <variable>
            <name>host</name>
            <value><host_name></value>
          </variable>
          <variable>
            <name>ipaddr</name>
            <value>1.2.3.4</value>
          </variable>
          <variable>
            <name>uptime</name>
            <value>5781</value>
          </variable>
        </key_ids>
      </names>
    </schemas>
```

```

</file>
<file>
  <number>2</number>
  <schemas>
    <schema>system</schema>
    <names>
      <name>system</name>
      <key_ids>
        <key_id>none</key_id>
        <variable>
          <name>host</name>
          <value><host_name></value>
        </variable>
        <variable>
          <name>ipaddr</name>
          <value>1.2.3.4</value>
        </variable>
        <variable>
          <name>uptime</name>
          <value>5781</value>
        </variable>
      </key_ids>
    </names>
  </schemas>
</file>
</bulkstats-operational>

```

## Exec CLI モデル

次の例では、Exec CLI モデルを使用しています。

### Curl を使用した「show version」出力の取得

Curl を使用して **show version** の出力を取得する例については、次を参照してください。

```

cat exec_cli_show_version.xml
<input><args>show version</args></input>
*****
[<user>@server] ]$ curl -u admin:pswd!
https://rtp-mitg-si06.cisco.com:234/api/running/staros_exec/_operations/exec --cert
/users/<user>/ssl_cert/client_cert/client.crt --key
/users/<user>/ssl_cert/client_cert/client.key --cacert
/users/<user>/ssl_cert/root_cert/rootCA.pem -X POST -T ./exec_cli_show_version.xml
<output xmlns='http://www.cisco.com/staros-exec'>
  <result>Active Software:
    Image Version:      21.2.M0.private
    Image Build Number: private
    Image Description:  Developer_Build
    Image Date:         Thu Feb 23 15:25:47 EST 2017
    Boot Image:         /flash/qvpc-si.bin.conf
    Source Commit ID:   bd234043a93c68873ea77444733a8c632356d161
  </result>
</output>

```

### Curl を使用した複数の show コマンド出力の取得

次に、コマンド間のデリミタとして「\r\n」を使用した **show build** および **show confdmgr** の出力を取得するための curl の使用例を示します。

```

cat exec_cli_show_build_and_confdmgr.xml
<input><args>show build \r\n show confdmgr</args></input>
*****
[<user>@server] ]$ curl -u admin:pswd!
https://rtp-mitg-si06.cisco.com:234/api/running/staros_exec/_operations/exec --cert
/users/<user>/ssl_cert/client_cert/client.crt --key
/users/<user>/ssl_cert/client_cert/client.key --cacert
/users/<user>/ssl_cert/root_cert/rootCA.pem -X POST -T ./
exec_cli_show_build_and_confdmgr.xml
<output xmlns='http://www.cisco.com/staros-exec'>
  <result>Active Software:
    Image Version:                21.2.M0.private
    Image Build Number:           private
    Image Description:            Developer_Build
    Image Date:                   Thu Feb 23 15:25:47 EST 2017
    Boot Image:                   /flash/qvpc-si.bin.confd
    Source Commit ID:             bd234043a93c68873ea77444733a8c632356d161
    Kernel Version:               2.6.38-staros-v3-ssi-64
    Kernel Machine Type:         x86_64

Build Information:
  Kernel Build:                  #1 SMP PREEMPT Wed Feb 22 12:28:49 EST 2017
  Image Build Type:              Production build
  Image Build User:              <user>
  Image Build Machine:          <host_name>
  Image Build Changeset Version: +
  Image Build Changeset Author: <user>
  Image Build Changeset Location: cisco.com
  Image Build Changeset Number:  bd234043a93c68873ea77444733a8c632356d161
  Image Build Changeset PID:    2017-02-23
*****
**** Local changes exist ****
*****

State Information
-----
State                Started
Subscriptions        5
Last successful id   1488-216669-170664
Last failed id      None
Username             Not configured
Bulkstats            Enabled
Event notification level Disabled
SNMP notifications  Disabled
REST interface authentication peer-fail
REST interface certificate rest-cert
REST interface host name Not configured

Interface            Status            Port
-----
NETCONF              Enabled           123
REST                  Enabled           234

Statistics
-----
Triggers              1
Replays               0
Notifications         27
Notification failures  0
Trigger failures      0
Replay failures        0
NETCONF notification failures 0
Unexpected failures    0
</result>

```

```
</output>
*****
```

## ConfD のアップグレードサポート

### ECS コマンド用の CLI ベースの YANG モデル

このリリースでは、**cisco-staros-cli-config.yang** モデルは、NSO を介して限定された一連の ECS (Enhanced Charging System) 設定コマンドをサポートします。

サウスバウンド側では、ConfD は、ConfD 管理エージェントによって提供される一連の API を介して呼び出された StarOS プロセスと通信します。ConfD CDB は、オブジェクトを保存するために ConfD によって使用されます。StarOS は、ConfD 提供の API を使用してデータベースにアクセスします。ConfD コンフィギュレーション データベースにデータが入力されると、StarOS は CLI アクセスによる全体的な設定の変更を引き続き許可します。結果として、CDB への自動更新は行われません。CDB は NETCONF インターフェイス経由でアップデートを受信します。CDB と StarOS コンフィギュレーション データベースを同期された状態に保つには、**cisco-staros-cli-config** YANG モデルでサポートされる設定オブジェクトに加えられた CLI アクセス (外部から NETCONF) を介したすべての変更を手動で CDB に適用する必要があります。

### CDB のシーディングと同期

**server confd** を有効にした後、StarOS で ConfD によって管理されているコンフィギュレーション データベース (CDB) のローカルコピーを使用して CDB を最初にシードする必要がある場合があります。シーディングの手順では、CLI ベースのすべての YANG モデルでサポートされている設定コマンドを含む StarOS プラットフォームで ConfD によって使用される CDB が作成されます。



#### 重要

- StarOS CLI を介して管理対象オブジェクトを手動で変更した場合は、次に説明する手順を繰り返して、実行時設定を NSO と再同期する必要があります。

- ステップ 1** Exec モードの **save configuration <url> confd** を実行して、ConfD でサポートされている StarOS 設定データを /flash デバイス上のファイルに保存します。
- ステップ 2** Exec モードの **show configuration error** を実行して、保存された設定を検証します。設定を適用する前にエラーを修正します。そうしないと、ConfD によって設定全体が拒否されます。
- ステップ 3** Exec モードの **configure confd <url>** を実行して、ConfD 設定を適用します。ConfD 設定が適用されると、デバイスは NSO 管理サービスへの NETCONF 接続を確立できるようになります。

**ステップ 4** デバイスを NSO と同期します。同期プロセスの詳細については、NSO のユーザーマニュアルを参照してください。

## show configuration confd コマンド

**confd** キーワードは **show configuration** コマンドの出力をフィルタリングして、CLI ベースの YANG モデルでサポートされている設定コマンドだけを表示します。

```
show configuration confd
```

サンプル出力は次のようになります。

```
[local]<host_name># show configuration confd
config
  context local
    server confd
    #exit
  active-charging service ecs
    ruledef rdl
      tcp any-match = TRUE
    #exit
  rulebase default
    #exit
#exit
end
[local]<host_name>#
```

## CDB のメンテナンス

ConfD コンフィギュレーションデータベース (CDB) のローカルコピーは、StarOS 上の ConfD によって管理されます。

すべての ConfD でサポートされている StarOS コンフィギュレーション コマンドを URL に表示して保存できます。そのため、**confd** キーワードが **show configuration** コマンドと **save configuration** コマンドに追加されました。

ConfD でサポートされている設定を URL に保存した後、Exec モードの **configure confd <url>** コマンドを使用して、それを CDB に直接適用できます。このコマンドは、*url* にあるファイルの内容を、ConfD の実行コンフィギュレーションに適用します。

上記のコマンドに関する詳細については、以下を参照してください。

## clear confdmgr confd cdb

この Exec モードコマンドは、ConfD が設定オブジェクトを保存するために使用する ConfD 設定データベース (CDB) の設定を消去します。StarOS は、ConfD が提供する API を使用してデータベースにアクセスします。



- (注) ConfD および NETCONF プロトコルのサポートを無効にするために、コンテキスト コンフィギュレーション モードの **no server confd** コマンドがローカルコンテキストで実行されている場合を除き、CDB を消去することはできません。

次に、CDB をクリアするためのコマンドシーケンスの例を示します。

```
[local]host_name# config
[local]host_name(config)# context local
[local]host_name(config-ctx)# no server confd
[local]host_name(config-ctx)# end
[local]host_name# clear confdmgr confd cdb
About to delete the ConfD configuration database
The running configuration is NOT affected.
Are you sure? [Yes|No]: y
[local]host_name#
```



- 注意 CDB のクリアは端末の動作です。その後、CDB を再設定する必要があります。

## configure confd <url>

この Exec モードのコマンドは、URL によって指定された設定スクリプトの内容を現在の ConfD 設定データベース (CDB) に適用します。

次に、コマンドシーケンスの例を示します。

```
[local]host_name# save configuration /flash/confd.config confd
[local]host_name# configure confd /flash/confd.config
Info: #!$$ StarOS V20.2 Chassis 52767e9ff9e207bed12c76f7f8a5352c
Info: config
Info:   active-charging service acs
Info:   rulebase default
Info:   #exit
Info:   #exit
Info: end
[local]host_name#
```

## save configuration <url> confd

キーワード **confd** が Exec モードの **save configuration** コマンドに追加されました。このキーワードは、YANG モデルでサポートされている設定コマンドのみを含むように、保存された設定コマンドをフィルタリングします。

次に、このプロセスのコマンドシンタックスを示します。

```
[local]host_name# save configuration <url> confd
```

設定コマンドの YANG モデルのサブセット出力は、**show file url <url>** コマンドを使用して表示できます。ここで、<url>は、設定を保存するために使用されるパス名です。保存された設定ファイルは、**configure confd** コマンドを使用して CDB に適用できます。

## サポートされている StarOS ECS 設定コマンド

このリリースでは、CLI ベースの YANG モデルで次の StarOS ECS コマンドがサポートされています。

- **ruledef <ruledef\_name>**
  - ip server-ip-address = \*
  - tcp-ether-port = \*
  - udp ether-port = \*
  - tcp ether-port-range = \*
  - udp ether-port range = \*
  - tcp-any-match = \*
  - udp any-match = \*
  - http url = \*
  - httpcookie = \*
  - http x-header = \*
- **group-of-ruledefs <ruledefs\_group\_name>**
  - add-ruledef priority = \*
- **qos-group-of-ruledefs <group\_name>**
  - add-group-of-ruledef <group\_of\_ruledef\_name>
- **charging-action <charging\_action\_name>**
  - flow-idle-timeout <seconds>
  - content-id l
  - service-identifier <service\_id>
  - billing-action egcdr
- **rulebase <rulebase\_name>**
  - action priority <priority\_number> group-of-ruledefs <ruledefs\_group\_name> charging-action <charging\_action\_name>



---

(注) "=" indicates support for every option following the prior keyword/value.

---





## 付録 **D**

# ICSR のチェックポイント

この付録では、シャージ間セッションリカバリのフレームワークによって使用されるマクロチェックポイントとマイクロチェックポイントについて説明します。チェックポイントは、サービス冗長性プロトコル (SRP) を介してアクティブとスタンバイの ICSR シャージの間で交換されます。

ここでは、次の内容について説明します。

- [チェックポイントの概要 \(497 ページ\)](#)
- [マクロチェックポイント \(498 ページ\)](#)
- [マイクロチェックポイント \(499 ページ\)](#)

## チェックポイントの概要

シャージ間セッションリカバリ (ICSR) には、ICSR フレームワーク内の `sessmgr` インスタンスレベルのチェックポイントのフレームワークが用意されています。チェックポイントは、アプリケーションのステータスのスナップショットです。チェックポイントは、`sessmgr` がピアシャージにインスタンスレベルの情報をプッシュするために使用できます。

インスタンスレベルのチェックポイントは、特定の `sessmgr` インスタンスにメッセージを送信します。GGSN、PDSN、P-GW、S-GW、または SGSN などの各アプリケーションがチェックポイントメッセージの符号化と復号を行います。ICSR フレームワークは、インスタンスレベルのチェックポイント情報と関連する統計情報を転送するための API を備えています。

マクロチェックポイントには完全なセッション情報が含まれており、マイクロチェックポイントにはいくつかの変数のみが含まれています。マクロチェックポイントは、電源投入時とリロード時にアクティブシャージからスタンバイシャージに最初に送信され、その後は定期的な送信されます。スタンバイシャージは、マクロチェックポイントを受信すると、そのセッションに関連する既存のすべての CRR (コールリカバリレコード) や CLP (コール回線ポイント) をクリアし、新しい CRR や CLP を作成します。マクロチェックポイントは、フルチェックポイント (FC) とも呼ばれます。

処理サイクルとメモリを節約するために、アクティブシャージからスタンバイシャージへの動的および定期的な更新はマイクロチェックポイントを使用して実行されます。

Exec モードの **show srp info** コマンドの出力には、SRP チェックポイントの完全なリストが表示されます。

## マクロチェックポイント

このセクションでは、ICSR マクロチェックポイントについて簡単に説明します。

### GGSN\_APN ID MAPPING

このマクロチェックポイントは、スタンバイシャーシ上の APN 名 をマッピングするために、アクティブシャーシからスタンバイシャーシに送信されます。

- 時間ベース：なし
- 頻度：該当なし
- イベントベース：対応
- イベント：sessmgr 間で TCP 接続が確立されるたびに発生し、READY\_STATE に移動します。
- アカウンティング：非対応
- 差分/累積：該当なし
- 関連 CLI コマンド：**show session subsystem facility sessmgr instance <instance no> debug-info** and **show srp micro-checkpoint statistics**

### インスタンスレベルのチェックポイント

このマクロチェックポイントは、新しいルールをスタンバイシャーシに送信するために、ECS (Enhanced Charging システム) によって生成されます。また、スタンバイシャーシのルールを削除または変更するために、ECS でも使用されます。

- 時間ベース：対応
- 頻度：30 分
- イベントベース：対応
- イベント：発生：
  1. 新しいルールがアクティブシャーシに追加または削除された場合。
  2. ECS が定期的にマイクロチェックポイントに登録されている場合は、30 分ごと。
- アカウンティング：—
- 差分/累積：—
- 関連 CLI コマンド：**show session subsystem facility sessmgr instance <instance no> debug-info** and **show srp micro-checkpoint statistics**

## SERVICE\_ID MAPPING

このマクロチェックポイントは、スタンバイシャーシのサービスIDをマッピングするために、アクティブシャーシからスタンバイシャーシに送信されます。

- 時間ベース：なし
- 頻度：該当なし
- イベントベース：対応
- イベント：sessmgr 間で TCP 接続が確立されるたびに発生し、READY\_STATE に移動します。
- アカウンティング：非対応
- 差分/累積：該当なし
- 関連 CLI コマンド：**show session subsystem facility sessmgr instance <instance no> debug-info**

## VPNMGR\_ID MAPPING

このマクロチェックポイントは、スタンバイシャーシ上の VPN をマッピングするために、アクティブシャーシからスタンバイシャーシに送信されます。

- 時間ベース：なし
- 頻度：該当なし
- イベントベース：対応
- イベント：sessmgr 間で TCP 接続が確立されるたびに発生し、READY\_STATE に移動します。
- アカウンティング：非対応
- 差分/累積：該当なし
- 関連 CLI コマンド：**show session subsystem facility sessmgr instance <instance no> debug-info**

## マイクロチェックポイント

このセクションでは、アプリケーションカテゴリ別のマイクロチェックポイントの特性について簡単に説明します。

マイクロチェックポイントは、次のカテゴリの下にアルファベット順に表示されます。

- [未分類 \(500 ページ\)](#)
- [DCCA カテゴリ \(501 ページ\)](#)
- [ECS カテゴリ \(501 ページ\)](#)
- [ePDG カテゴリ \(505 ページ\)](#)
- [ファイアウォール/ECS カテゴリ \(506 ページ\)](#)
- [GGSN カテゴリ \(507 ページ\)](#)
- [Gx インターフェイスカテゴリ \(509 ページ\)](#)
- [NAT カテゴリ \(509 ページ\)](#)
- [P-GW カテゴリ \(512 ページ\)](#)

- Rf インターフェイスカテゴリ (514 ページ)
- S6b インターフェイスカテゴリ (515 ページ)
- SaMOG カテゴリ (515 ページ)

## 未分類

### SESS\_UCHKPT\_CMD\_INVALIDATE\_CRR

このマイクロチェックポイントは、削除されたコールをクリアするためにスタンバイシャーシに送信されます。スタンバイシャーシで削除する必要があるコール ID とその他の情報を伝送します。

- 時間ベース：なし
- 頻度：該当なし
- イベントベース：対応
- イベント：アクティブシャーシでコールが削除されたときに発生します。
- アカウンティング：非対応
- 差分/累積：該当なし
- **CMD-ID**：1
- 関連 CLI コマンド：なし

### SESS\_UCKKPT\_CMD\_UPDATE\_CLPSTATS

このマイクロチェックポイントは、VoLTE データの統計情報を送信します。

- 時間ベース：対応
- 頻度：—
- イベントベース：対応
- イベント：ICSR バックグラウンドチェックポイント中に発生します。シャーシのスイッチオーバーにより、VoLTE データ統計情報の送信がトリガーされます。
- アカウンティング：—
- 差分/累積：—
- **CMD-ID**：4
- 関連 CLI コマンド：なし

### SESS\_UCHKPT\_CMD\_UPDATE\_IDLESECS

このマイクロチェックポイントは、アイドルタイムアウトまでの残りの秒数を送信します。

- 時間ベース：対応
- 頻度：—
- イベントベース：非対応
- イベント：ICSR バックグラウンドチェックポイント中に発生します。
- アカウンティング：非対応
- 差分/累積：該当なし

- **CMD-ID** : 2
- **関連 CLI コマンド** : なし

## DCCA カテゴリ

### SESS\_UCHKPT\_CMD\_DCCA\_SESS\_INFO

このマイクロチェックポイントは、クレジット制御（CC）関連の情報を送信します。

- **時間ベース** : 対応
- **頻度** : GR マイクロチェックポイントの場合は 18 秒
- **イベントベース** : 対応
- **イベント** : マイクロチェックポイント/CCA/正を想定の状態遷移とともに送信
- **アカウンティング** : 対応
- **差分/累積** : 累積
- **CMD-ID** : 19
- **関連 CLI コマンド** : なし

## ECS カテゴリ

### SESS\_UCHKPT\_CMD\_ACS\_CALL\_INFO

このマイクロチェックポイントは、重要な ECS コールレベルのデータを送信します。

- **時間ベース** : 対応
- **頻度** : —
- **イベントベース** : 対応
- **イベント** : ECS のコールレベル情報が作成または変更されるたびに発生します。
- **アカウンティング** : 非対応
- **差分/累積** : 該当なし
- **CMD-ID** : 179
- **関連 CLI コマンド** : なし

### SESS\_UCHKPT\_CMD\_ACS\_GX\_LI\_INFO

このマイクロチェックポイント ソースは、ECS によって維持される合法的傍受（LI）関連情報です。

- **時間ベース** : 対応
- **頻度** : —
- **イベントベース** : 対応
- **イベント** : LI 情報が作成または変更されるたびに発生します。
- **アカウンティング** : 非対応
- **差分/累積** : 該当なし

- **CMD-ID** : 75
- **関連 CLI コマンド** : なし

## SESS\_UCHKPT\_CMD\_ACS\_SESS\_INFO

このマイクロチェックポイントは、ECS レベルのベアラー関連データを送信します。

- **時間ベース** : 対応
- **頻度** : —
- **イベントベース** : 対応
- **イベント** : ECS ベアラー情報が作成または変更されるたびに発生します。
- **アカウンティング** : 非対応
- **差分/累積** : 該当なし
- **CMD-ID** : 33
- **関連 CLI コマンド** : なし

## SESS\_UCHKPT\_CMD\_DEL\_ACS\_CALL\_INFO

このマイクロチェックポイントは、リリースコールのイベントが発生したことを通知します。

- **時間ベース** : なし
- **頻度** : 該当なし
- **イベントベース** : 対応
- **イベント** : ECS リリースコールのメッセージが処理されるたびに発生します。
- **アカウンティング** : 非対応
- **差分/累積** : 該当なし
- **CMD-ID** : 188
- **関連 CLI コマンド** : —

## SESS\_UCHKPT\_CMD\_DEL\_ACS\_SESS\_INFO

このマイクロチェックポイントは、ベアラーのリリースイベントが発生したことを通知します。

- **時間ベース** : なし
- **頻度** : 該当なし
- **イベントベース** : 対応
- **イベント** : ECS のベアラーのリリースメッセージが処理されるたびに発生します。
- **アカウンティング** : 非対応
- **差分/累積** : 該当なし
- **CMD-ID** : 187
- **関連 CLI コマンド** : なし

## SESS\_UCHKPT\_CMD\_DYNAMIC\_CHRG\_CA\_INFO

このマイクロチェックポイントは、ECSによって維持されるダイナミックチャージングのアクション情報を送信します。

- 時間ベース：対応
- 頻度：—
- イベントベース：対応
- イベント：ダイナミックチャージングのアクション情報が作成または変更されるたびに発生します。
- アカウンティング：非対応
- 差分/累積：該当なし
- **CMD-ID**：141
- 関連 CLI コマンド：なし

## SESS\_UCHKPT\_CMD\_DYNAMIC\_CHRG\_DEL\_CA\_INFO

このマイクロチェックポイントは、ダイナミックチャージングのアクションが削除されていることを通知します。

- 時間ベース：なし
- 頻度：該当なし
- イベントベース：対応
- イベント：ダイナミックチャージングのアクションが削除されるたびに発生します。
- アカウンティング：非対応
- 差分/累積：該当なし
- **CMD-ID**：183
- 関連 CLI コマンド：なし

## SESS\_UCHKPT\_CMD\_DYNAMIC\_CHRG\_DEL\_QG\_INFO

このマイクロチェックポイントは、ダイナミック QoS グループが削除されたことを通知します。

- 時間ベース：なし
- 頻度：該当なし
- イベントベース：対応
- イベント：ダイナミック QoS グループが削除されるたびに発生します。
- アカウンティング：非対応
- 差分/累積：該当なし
- **CMD-ID**：182
- 関連 CLI コマンド：なし

## SESS\_UCHKPT\_CMD\_DYNAMIC\_CHRG\_QG\_INFO

このマイクロチェックポイントは、ECS によって維持されるダイナミック QoS グループ関連情報を送信します。

- 時間ベース：対応
- 頻度：—
- イベントベース：対応
- イベント：ダイナミック QoS グループ情報が作成または変更されるたびに発生します。
- アカウンティング：非対応
- 差分/累積：該当なし
- **CMD-ID**：140
- 関連 CLI コマンド：なし

## SESS\_UCHKPT\_CMD\_DYNAMIC\_RULE\_DEL\_INFO

このマイクロチェックポイントは、ダイナミックルールが削除されていることを通知します。

- 時間ベース：なし
- 頻度：—
- イベントベース：対応
- イベント：ダイナミックルールが削除されるたびに発生します。
- アカウンティング：非対応
- 差分/累積：該当なし
- **CMD-ID**：178
- 関連 CLI コマンド：なし

## SESS\_UCHKPT\_CMD\_DYNAMIC\_RULE\_INFO

このマイクロチェックポイントソースは、ECS によって維持される事前定義済みのダイナミックルール関連情報です。

- 時間ベース：対応
- 頻度：—
- イベントベース：対応
- イベント：ダイナミックルールが作成または変更されるたびに発生します。
- アカウンティング：非対応
- 差分/累積：該当なし
- **CMD-ID**：43
- 関連 CLI コマンド：なし



## ePDG カテゴリ

### SESS\_UCHKPT\_CMD\_DELETE\_EPDG\_BEARER

このマイクロチェックポイントは、アクティブシャーシとスタンバイシャーシ間で削除された ePDG ベアラーを同期します。

- 時間ベース：なし
- 頻度：該当なし
- イベントベース：対応
- イベント：該当なし
- アカウンティング：対応
- 差分/累積：累積
- **CMD-ID**：110
- 関連する CLI コマンド：show srp micro-checkpoint statistics debug-info

### SESS\_UCHKPT\_CMD\_UPDATE\_EPDG\_BEARER

このマイクロチェックポイントは、アクティブシャーシとスタンバイシャーシ間で ePDG ベアラーを同期します。

- 時間ベース：なし
- 頻度：該当なし
- イベントベース：非対応
- イベント：該当なし
- アカウンティング：対応
- 差分/累積：累積
- **CMD-ID**：110
- 関連する CLI コマンド：show srp micro-checkpoint statistics debug-info

### SESS\_UCHKPT\_CMD\_UPDATE\_EPDG\_PEER\_ADDR

このマイクロチェックポイントは、アクティブシャーシとスタンバイシャーシ間で ePDG ピアアドレスを同期します。

- 時間ベース：なし
- 頻度：該当なし
- イベントベース：対応
- イベント：—
- アカウンティング：対応
- 差分/累積：累積
- **CMD-ID**：110
- 関連する CLI コマンド：show srp micro-checkpoint statistics debug-info

## SESS\_UCHKPT\_CMD\_UPDATE\_EPDG\_REKEY

このマイクロチェックポイントは、アクティブシャーシとスタンバイシャーシ間で ePDG キー生成統計情報を同期します。

- 時間ベース：対応
- 頻度：30 秒
- イベントベース：非対応
- イベント：該当なし
- アカウンティング：対応
- 差分/累積：累積
- **CMD-ID**：110
- 関連する CLI コマンド：show srp micro-checkpoint statistics debug-info

## SESS\_UCHKPT\_CMD\_UPDATE\_EPDG\_STATS

このマイクロチェックポイントは、アクティブシャーシとスタンバイシャーシ間のセッション統計情報を同期します。

- 時間ベース：対応
- 頻度：30 秒
- イベントベース：非対応
- イベント：該当なし
- アカウンティング：対応
- 差分/累積：累積
- **CMD-ID**：110
- 関連する CLI コマンド：show srp micro-checkpoint statistics debug-info

## ファイアウォール/ECS カテゴリ

### SESS\_UCHKPT\_CMD\_SFW\_DEL\_RULE\_INFO

このマイクロチェックポイントは、ベアラの ruledef が削除されると送信されます。

- 時間ベース：なし
- 頻度：該当なし
- イベントベース：対応
- イベント：PCRF が事前定義されたステートフルファイアウォールのアクセスルールを無効にするコマンドを送信するたびに発生します。
- アカウンティング：非対応
- 差分/累積：該当なし
- **CMD-ID**：186
- 関連 CLI コマンド：なし

## SESS\_UCHKPT\_CMD\_SFW\_RULE\_INFO

このマイクロチェックポイントは、動的に有効になったステートフルファイアウォール (SFW) のアクセスルールの追加を通知します。

- 時間ベース：なし
- 頻度：該当なし
- イベントベース：対応
- イベント：PCRF が事前定義された SFW アクセスルールを有効にするコマンドを送信するたびに発生します。
- アカウンティング：対応
- 差分/累積：累積
- **CMD-ID**：185
- 関連 CLI コマンド：なし

## GGSN カテゴリ

### SESS\_UCHKPT\_CMD\_GGSN\_DELETE\_SUB\_SESS

このマイクロチェックポイントは、セカンダリベアラーが削除されたときに更新を送信します。

- 時間ベース：なし
- 頻度：該当なし
- イベントベース：対応
- イベント：セカンダリベアラーの削除時に発生
- アカウンティング：—
- 差分/累積：—
- **CMD-ID**：117
- 関連 CLI コマンド：なし

### SESS\_UCHKPT\_CMD\_GGSN\_UPDATE\_RPR

RPR（復元力のあるパケットリング）が GGSN サービス内に設定されている場合は、セカンダリベアラーの作成時に RPR タイマーが起動されます。このチェックポイントは、このタイマーの有効期限が切れたときに送信されます。

- 時間ベース：対応
- 頻度：RPR タイマー
- イベントベース：対応
- イベント：セカンダリベアラー作成の RPR タイマーが期限切れになると発生します。
- アカウンティング：—
- 差分/累積：—
- **CMD-ID**：118
- 関連 CLI コマンド：—

## SESS\_UCHKPT\_CMD\_GGSN\_UPDATE\_SESSION

このマイクロチェックポイントは、次のシナリオとなる更新を除き、ネットワークまたは UE によって開始された更新手順で送信されます。

- ベアラの作成または削除
- TFT の変更または RAT 間 ハンドオーバー
- Gn-Gp ハンドオフ

このマイクロチェックポイントに関連付けられているパラメータを次に示します。

- 時間ベース：なし
- 頻度：該当なし
- イベントベース：対応
- イベント：更新がネットワークで開始されるか、または UE で開始されたときに発生します。
- アカウンティング：非対応
- 差分/累積：該当なし
- **CMD-ID**：171
- 関連 CLI コマンド：`show srp checkpoint statistics active verbose` と `show session subsystem facility sessmgr instance <instance_number> debug-info`.

## SESS\_UCHKPT\_CMD\_GGSN\_UPDATE\_STATS

このマイクロチェックポイントは、セッションの統計情報を定期的に送信します。

- 時間ベース：対応
- 頻度：5 分ごと
- イベントベース：非対応
- イベント：該当なし
- アカウンティング：対応
- 差分/累積：累積
- **CMD-ID**：116
- 関連 CLI コマンド：なし

## SESS\_UCHKPT\_CMD\_UPDATE\_COA\_PARAMS

このマイクロチェックポイントは、入力および出力 ACL パラメータを更新します。

- 時間ベース：—
- 頻度：—
- イベントベース：対応
- イベント：COA（認可変更）の応答
- アカウンティング：—
- 差分/累積：—
- **CMD-ID**：83
- 関連 CLI コマンド：なし

## Gx インターフェイスカテゴリ

### SESS\_UCHKPT\_CMD\_ACS\_VOLUME\_USAGE

このマイクロチェックポイントは、Gx アカウンティングバケットを介してボリューム使用率を送信します。

- 時間ベース：対応
- 頻度：aamgr マイクロチェックポイントの場合は 4 秒、GR マイクロチェックポイントの場合は 18 秒
- イベントベース：非対応
- イベント：マクロチェックポイントとともに送信
- アカウンティング：対応
- 差分/累積：累積
- CMD-ID：79
- 関連 CLI コマンド：— なし

### SESS\_UCHKPT\_CMD\_UPDATE\_SGX\_INFO

このマイクロチェックポイントは、Gx セッション関連の情報を送信します。

- 時間ベース：なし
- 頻度：該当なし
- イベントベース：対応
- イベント：PCRF からの CCA-I/U または RAR の受信時にトリガーされます。
- アカウンティング：対応
- 差分/累積：累積
- CMD-ID：137
- 関連 CLI コマンド：なし

## NAT カテゴリ

### SESS\_UCHKPT\_CMD\_GR\_UPDATE\_NAT\_REALM\_PORT\_INFO1

このマイクロチェックポイントは、他のサブスクリイバと NAT IP アドレスを共有しているサブスクリイバに対して、ポートチャックが割り当てられるか、または割り当て解除されるときに送信されます。ポートチャックは、そのサブスクリイバのデータを受信している間、割り当てられるか、割り当てが解除されます。

- 時間ベース：なし
- 頻度：該当なし
- イベントベース：対応
- イベント：新しい NAT ポートチャックが割り当てられるか、または削除されるとトリガーされます。
- アカウンティング：非対応

- 差分/累積：該当なし
- **CMD-ID**：105
- 関連 CLI コマンド：なし

## SESS\_UCHKPT\_CMD\_GR\_UPDATE\_NAT\_REALMS

このマイクロチェックポイントは、NAT IP サブスクリバから解除されたときに送信されません。

オンデマンドの場合は、特定の NAT レルムに一致する最初のパケットが受信され、NAT IP アドレスがサブスクリバに割り当てられるとトリガーされます。

これがオンデマンドでない場合は、コールセットアップ時に NAT IP アドレスが割り当てられて、このマイクロチェックポイントが送信されます。

- 時間ベース：なし
- 頻度：該当なし
- イベントベース：対応
- イベント：NAT IP アドレスがサブスクリバに割り当てられたか、またはその割当をサブスクリバから解除されたときにトリガーされます。
- アカウンティング：非対応
- 差分/累積：該当なし
- **CMD-ID**：45
- 関連 CLI コマンド：なし

## SESS\_UCHKPT\_CMD\_NAT\_SIP\_ALG\_CALL\_INFO

このマイクロチェックポイントは、新しい SIP フローがサブスクリバに対して作成または削除されたときに送信されます（SIP データはサブスクリバ経由で渡されます）。

- 時間ベース：なし
- 頻度：該当なし
- イベントベース：対応
- イベント：新しい SIP フローが作成または削除されたときにトリガーされます。
- アカウンティング：非対応
- 差分/累積：該当なし
- **CMD-ID**：98
- 関連 CLI コマンド：なし

## SESS\_UCHKPT\_CMD\_NAT\_SIP\_ALG\_CONTACT\_PH\_INFO

このマイクロチェックポイントは、受信した SIP パケットが分析され、ピンホールが NAT ファイアウォールで作成されたときに送信されます。

- 時間ベース：なし
- 頻度：該当なし
- イベントベース：対応

- イベント：SIP パケットが NAT ファイアウォールでピンホールを作成するときにトリガーされます。
- アカウンティング：非対応
- 差分/累積：該当なし
- **CMD-ID**：97
- 関連 CLI コマンド：なし

## SESS\_UCHKPT\_CMD\_UPDATE\_DSK\_FLOW\_CHKPT\_INFO

このマイクロチェックポイントは、新しい NAT フローがサブスクリバに対して作成または削除されたときに送信されます（データはサブスクリバ経由で渡されます）。

このチェックポイントはタイマーから送信されますが、タイマーベースではありません。タイマーは、タイマーが起動するたびに（粒度は 2 秒）調整のために使用されます（10 マイクロチェックポイント）。これは、マイクロチェックポイントを必要とする新しいフローがある場合にのみ送信されます。それ以外の場合は、micro-micro-checkpoints は送信されません。

- 時間ベース：なし
- 頻度：上記の説明を参照してください。
- イベントベース：対応
- イベント：新しい NAT フローが作成または削除されたときにトリガーされます。
- アカウンティング：非対応
- 差分/累積：該当なし
- **CMD-ID**：96
- 関連 CLI コマンド：なし

## SESS\_UCHKPT\_CMD\_UPDATE\_NAT\_BYPASS\_FLOW\_INFO

このマイクロチェックポイントは、サブスクリバに対して NAT が有効になっているが、（ルール的一致に基づいて）このフローに対してバイパス NAT（NAT なし）が設定されている場合に送信され、新しいバイパスフローが作成されます。

このチェックポイントは、フローが追加と削除の両方が行われると送信されます。

- 時間ベース：なし
- 頻度：該当なし
- イベントベース：対応
- イベント：バイパス NAT が有効な新しいフローが作成または削除されたときにトリガーされます。
- アカウンティング：非対応
- 差分/累積：該当なし
- **CMD-ID**：60
- 関連 CLI コマンド：なし

## P-GW カテゴリ

### SESS\_UCHKPT\_CMD\_PGW\_DELETE\_SUB\_SESS

将来的な使用のために予約されています。

### SESS\_UCHKPT\_CMD\_PGW\_OVRCHRG\_PRTCTN\_INFO

このマイクロチェックポイントは、S-GW が MBR 内の過剰課金保護ビットを設定していることを示します。

- 時間ベース：なし
- 頻度：該当なし
- イベントベース：対応
- イベント：S-GW が過剰課金保護ビットを超えて設定した場合にトリガーされます。
- アカウンティング：非対応
- 差分/累積：該当なし
- CMD-ID：159
- 関連 CLI コマンド：なし

### SESS\_UCHKPT\_CMD\_PGW\_SGWRESTORATION\_INFO

このマイクロチェックポイントは、S-GW がダウンしているときにコールがアップ状態を維持する間隔を示します。

- 時間ベース：なし
- 頻度：該当なし
- イベントベース：対応
- イベント：S-GW が復元モードになったときにトリガーされます。
- アカウンティング：非対応
- 差分/累積：該当なし
- CMD-ID：158
- 関連 CLI コマンド：なし

### SESS\_UCHKPT\_CMD\_PGW\_UBR\_MBR\_INFO

このマイクロチェックポイントは、UBR/MBR の手順によって次のシナリオが発生した場合を除き、UBR (Update Bearer Request) または MBR (Modify Bearer Request) の最後に送信されます。

- TFT の変更
- 非表示コールのベアラー更新または変更
- 純粋な P から非表示、または非表示から純粋な P に変更
- テクノロジー間ハンドオフ (たとえば、WiFi から LTE へ)

このマイクロチェックポイントに関連付けられているパラメータを次に示します。



- 時間ベース：なし
- 頻度：該当なし
- イベントベース：対応
- イベント：UBR または MBR プロシージャの結果として発生します。
- アカウンティング：非対応
- 差分/累積：該当なし
- **CMD-ID**：193
- 関連する CLI コマンド：**show srp checkpoint statistics active verbose** および **show session subsystem facility sessmgr instance < instance\_number > debug-info**

## SESS\_UCHKPT\_CMD\_PGW\_UPDATE\_APN\_AMBR

将来的な使用のために予約されています。

## SESS\_UCHKPT\_CMD\_PGW\_UPDATE\_INFO

将来的な使用のために予約されています。

## SESS\_UCHKPT\_CMD\_PGW\_UPDATE\_LI\_PARAM

このマイクロチェックポイントは、このコールの合法的傍受 (LI) の状態を示します。

- 時間ベース：なし
- 頻度：該当なし
- イベントベース：対応
- イベント：このコールの LI 状態が変更された場合にトリガーされます。
- アカウンティング：非対応
- 差分/累積：該当なし
- **CMD-ID**：151
- 関連 CLI コマンド：なし

## SESS\_UCHKPT\_CMD\_PGW\_UPDATE\_PDN\_COMMON\_PARAM

将来的な使用のために予約されています。

## SESS\_UCHKPT\_CMD\_PGW\_UPDATE\_QOS

将来的な使用のために予約されています。

## SESS\_UCHKPT\_CMD\_PGW\_UPDATE\_SGW\_CHANGE

将来的な使用のために予約されています。

## SESS\_UCHKPT\_CMD\_PGW\_UPDATE\_STATS

このマイクロチェックポイントは、セッションの統計情報を定期的に送信します。

- 時間ベース：対応

- 頻度：5 分ごと
- イベントベース：非対応
- イベント：該当なし
- アカウンティング：対応
- 差分/累積：累積
- **CMD-ID**：65
- 関連 CLI コマンド：なし

## Rf インターフェイスカテゴリ

### SESS\_UCHKPT\_CMD\_ACS\_ACCOUNTING\_TYPE\_QCI\_RF

このマイクロチェックポイントは、SDF+QCI-based Rf アカウンティングバケットの変更を示します。

- 時間ベース：対応
- 頻度：aamgr チェックポイントの場合は 4 秒、GR チェックポイントの場合は 18 秒
- イベントベース：非対応
- イベント：該当なし
- アカウンティング：対応
- 差分/累積：累積
- **CMD-ID**：126
- 関連 CLI コマンド：なし

### SESS\_UCHKPT\_CMD\_ACS\_ACCOUNTING\_TYPE\_QCI\_RF\_WITH\_FC

このマイクロチェックポイントは、完全な SDF+QCI-based Rf アカウンティングバケットを示します。

- 時間ベース：対応
- 頻度：aamgr チェックポイントの場合は 4 秒、GR チェックポイントの場合は 18 秒
- イベントベース：非対応
- イベント：マイクロチェックポイントとともに送信されます。
- アカウンティング：対応
- 差分/累積：累積
- **CMD-ID**：164
- 関連 CLI コマンド：なし

### SESS\_UCHKPT\_CMD\_ACS\_ACCOUNTING\_TYPE\_RATING\_GROUP\_RF

このマイクロチェックポイントは、SDF ベースの Rf アカウンティングバケットの変更を示します。

- 時間ベース：対応
- 頻度：aamgr チェックポイントの場合は 4 秒、GR チェックポイントの場合は 18 秒

- イベントベース：非対応
- イベント：該当なし
- アカウンティング：対応
- 差分/累積：累積
- **CMD-ID**：125
- 関連 CLI コマンド：なし

## SESS\_UCHKPT\_CMD\_ACS\_ACCOUNTING\_TYPE\_RATING\_GROUP\_RF\_WITH\_FC

このマイクロチェックポイントは、完全な SDF ベースの Rf アカウンティングバケットを示します。

- 時間ベース：対応
- 頻度：aamgr チェックポイントの場合は 4 秒、GR チェックポイントの場合は 18 秒
- イベントベース：非対応
- イベント：マクロチェックポイントとともに送信されます。
- アカウンティング：対応
- 差分/累積：累積
- **CMD-ID**：163
- 関連 CLI コマンド：なし

## S6b インターフェイスカテゴリ

### SESS\_UCHKPT\_CMD\_S6B\_INFO

このマイクロチェックポイントは、S6b インターフェイスを介して再許可が発生したときに復元プライオリティインジケータを送信します。

- 時間ベース：なし
- 頻度：該当なし
- イベントベース：対応
- イベント：S6b 再許可によって、復元プライオリティインジケータの値が変更されると発生します。
- アカウンティング：非対応
- 差分/累積：該当なし
- **CMD-ID**：202
- 関連 CLI コマンド：なし

## SaMOG カテゴリ

### SESS\_UCHKPT\_CMD\_CGW\_DELETE\_BEARER

将来的な使用のために予約されています。

## SESS\_UCHKPT\_CMD\_CGW\_DELETE\_PDN

このマイクロチェックポイントは、PDN 接続が削除されたことを示します。

- 時間ベース：なし
- 頻度：該当なし
- イベントベース：対応
- イベント：SaMOG が Delete-Session-Req を送信したとき、または Delete-Bearer-Request を受信したときに発生します。
- アカウンティング：非対応
- 差分/累積：該当なし
- CMD-ID：169
- 関連 CLI コマンド：show subscriber samog-only full

## SESS\_UCHKPT\_CMD\_CGW\_UPDATE\_BEARER\_QOS

このマイクロチェックポイントは、ベアラーの QoS 更新を示します。

- 時間ベース：なし
- 頻度：該当なし
- イベントベース：対応
- イベント：再認証（AAA サーバーから受信した AAR）または Update-Bearer-Request によるベアラー QoS の変更を P-GW から受信したときに発生します。
- アカウンティング：非対応
- 差分/累積：該当なし
- CMD-ID：167
- 関連 CLI コマンド：show subscriber samog-only full

## SESS\_UCHKPT\_CMD\_CGW\_UPDATE\_PDN

このマイクロチェックポイントは、APN-AMBR の変更に対する PDN 更新を示します。

- 時間ベース：なし
- 頻度：該当なし
- イベントベース：対応
- イベント：再許可（AAA サーバーから受信した AAR）または Update-Bearer-Request により、APN-AMBR の変更が P-GW から受信された場合に発生します。
- アカウンティング：非対応
- 差分/累積：該当なし
- CMD-ID：168
- 関連 CLI コマンド：show subscriber samog-only full

## SESS\_UCHKPT\_CMD\_CGW\_UPDATE\_STATS

将来的な使用のために予約されています。

## SESS\_UCHKPT\_CMD\_CGW\_UPDATE\_UE\_PARAM

将来的な使用のために予約されています。

## SESS\_UCHKPT\_CMD\_SAMOG\_ACCT\_INTERIM\_INFO

このマイクロチェックポイントは、WLC からアカウントिंग要求 (INTERIM-UPDATE) を受信すると、SaMOG セッションに対して送信されます。

- 時間ベース：なし
- 頻度：該当なし
- イベントベース：対応
- イベント：WLC からアカウントिंग要求 (INTERIM-UPDATE) を受信したときに発生します。
- アカウントिंग：非対応
- 差分/累積：該当なし
- **CMD-ID**：177
- 関連 CLI コマンド：**show subscriber samog-only full**

## SESS\_UCHKPT\_CMD\_SAMOG\_ACCT\_START\_INFO

このマイクロチェックポイントは、WLC (ワイヤレス LAN コントローラ) から Accounting Req (START) を受信した場合に、SaMOG セッション用に送信されます。

- 時間ベース：なし
- 頻度：該当なし
- イベントベース：対応
- イベント：WLC から Account Req (STOP) を受信した場合に発生します。
- アカウントिंग：非対応
- 差分/累積：該当なし
- **CMD-ID**：174
- 関連 CLI コマンド：**show subscriber samog-only full**

## SESS\_UCHKPT\_CMD\_SAMOG\_EOGR\_TUNNEL\_INFO

このマイクロチェックポイントは、EoGRE サブスクライバセッションの RG 間ハンドオフ用に送信されます。このチェックポイントは、VMAC アドレスと WLC EoGRE トンネルのエンドポイントアドレスを更新します。

- 時間ベース：なし
- 頻度：該当なし
- イベントベース：対応
- イベント：DHCP 検出メッセージが別の EoGRE トンネルを介して受信されるたびに発生します。
- アカウントिंग：非対応
- 差分/累積：該当なし

- **CMD-ID** : 201
- **関連 CLI コマンド** : `show subscriber samog-only full`

## SESS\_UCHKPT\_CMD\_SAMOG\_GTPV1\_UPDATE\_PDN\_INFO

このマイクロチェックポイントは、PDN 情報を更新するため、GGSN からの Update-PDP-Context-Req の受信時に SaMOG セッションに対して送信されます。

- **時間ベース** : なし
- **頻度** : 該当なし
- **イベントベース** : 対応
- **イベント** : GGSN からの Update-PDP-Context-Req の SaMOG 処理が成功した後に発生します。
- **アカウントティング** : 非対応
- **差分/累積** : 該当なし
- **CMD-ID** : 191
- **関連 CLI コマンド** : `show subscriber samog-only full`

## SESS\_UCHKPT\_CMD\_SAMOG\_HANDOFF\_AUTHEN\_INFO

このマイクロチェックポイントは、サブスクライバセッションがハンドオフ状態のときにサブスクライバを再認証する SaMOG セッションに対して送信されます。

- **時間ベース** : なし
- **頻度** : 該当なし
- **イベントベース** : 対応
- **イベント** : 現在、ハンドオフ状態にある既存の SaMOG サブスクライバセッションの再認証が完了したときに発生します。
- **アカウントティング** : 非対応
- **差分/累積** : 該当なし
- **CMD-ID** : 176
- **関連 CLI コマンド** : `show subscriber samog-only full`

## SESS\_UCHKPT\_CMD\_SAMOG\_HANDOFF\_INIT\_INFO

このマイクロチェックポイントは、WLC (ワイヤレス LAN コントローラ) から Accounting Req (STOP) を受信した場合に、SaMOG セッション用に送信されます。

SaMOG は、サブスクライバからの Accounting Req (START) を予測しているため、ハンドオフが遅延します。

- **時間ベース** : なし
- **頻度** : 該当なし
- **イベントベース** : 対応
- **イベント** : WLC から Account Req (STOP) を受信した場合に発生します。
- **アカウントティング** : 非対応

- 差分/累積：該当なし
- **CMD-ID**：175
- 関連 CLI コマンド：**show subscriber samog-only full**

## SESS\_UCHKPT\_CMD\_SAMOG\_LI\_PROV\_INFO

このマイクロチェックポイントは、合法的傍受（LI）の Active-Camp-on モードにある SaMOG セッションに対して送信されます。

- 時間ベース：なし
- 頻度：該当なし
- イベントベース：対応
- イベント：SaMOG セッションの作成後で、LI トリガーを受信した後に発生します。
- アカウンティング：非対応
- 差分/累積：該当なし
- **CMD-ID**：189
- 関連 CLI コマンド：**show subscriber samog-only full**

## SESS\_UCHKPT\_CMD\_SAMOG\_MIPV6\_TIMER\_INFO

このマイクロチェックポイントは、SaMOG セッションのバインディング キャッシュ ライフ タイマーと MIPv6 バインディングステータスを更新します。

- 時間ベース：なし
- 頻度：該当なし
- イベントベース：対応
- イベント：ライフタイムがゼロの PMIPv6 PBU を WLC から受信するたびに発生します。
- アカウンティング：非対応
- 差分/累積：該当なし
- **CMD-ID**：190
- 関連 CLI コマンド：**show subscriber samog-only full**

## SESS\_UCHKPT\_CMD\_SAMOG\_MULTI\_ROUND\_AUTHEN\_INFO

このマイクロチェックポイントは、サブスクライバセッションの再認証中に、SaMOG がアクセスチャレンジを送信した後に、その時点で待機している場合に、SaMOG セッション用に送信されます。

- 時間ベース：なし
- 頻度：該当なし
- イベントベース：対応
- イベント：再認証中に、SaMOG が既存の SaMOG サブスクライバセッションのアクセスチャレンジを送信した後に発生します。
- アカウンティング：非対応
- 差分/累積：該当なし
- **CMD-ID**：184

- 関連 CLI コマンド : **show subscriber samog-only full**

## SESS\_UCHKPT\_CMD\_SAMOG\_REAUTHEN\_INFO

このマイクロチェックポイントは、サブスクリバの再認証が完了したときに、SaMOG セッション用に送信されます。

- 時間ベース : なし
- 頻度 : 該当なし
- イベントベース : 対応
- イベント : 既存の SaMOG サブスクリバセッションの再認証の完了時に発生します。
- アカウンティング : 非対応
- 差分/累積 : 該当なし
- **CMD-ID** : 172
- 関連 CLI コマンド : **show subscriber samog-only full**

## SESS\_UCHKPT\_CMD\_SAMOG\_REAUTHOR\_INFO

このマイクロチェックポイントは、サブスクリバの再許可が完了したときに、SaMOG セッション用に送信されます。

- 時間ベース : なし
- 頻度 : 該当なし
- イベントベース : 対応
- イベント : サブスクリバを再許可するために、AAA サーバーから AAR を受信して正常に処理するときに発生します。
- アカウンティング : 非対応
- 差分/累積 : 該当なし
- **CMD-ID** : 173
- 関連 CLI コマンド : **show subscriber samog-only full**





## 付録 E

# VPC-SI SDR CLI コマンドストリング

この付録では、グローバル コンフィギュレーション モードで **support record section** コマンドを使用して、レコードセクションに入力できる StarOS CLI コマンド文字列を示します。文字列が認識されるようにするには、二重引用符 ("" ) で囲む必要があります。この表は、デフォルトの文字列とデフォルト以外の文字列も示しています。

コマンド文字列の詳細については、『*Command Line Interface Reference*』またはコマンドのオンラインヘルプを参照してください。

この表は、デフォルトの文字列とデフォルト以外の文字列も示しています。これには、**Show support collection definitions** コマンドの出力シーケンスが反映されます。

表 38: VPC-SI SDR CLI コマンドストリング

番号	デフォルト SDR	コマンド文字列
0	有効	"show version verbose"
1	有効	"show clock"
2	有効	"show clock universal"
3	有効	"show configuration"
4	有効	"show_profile"
5	有効	"show context"
6	有効	"show boot"
7	有効	"show boot initial-config"
8	有効	"show system uptime"
9	無効	"show license information"
10	無効	"show license history"
11	無効	"show hardware inventory"
12	無効	"show hardware version"

番号	デフォルト SDR	コマンド文字列
13	無効	"show card hardware"
25	無効	"show card info"
26	有効	"show card diag"
27	有効	"show card table all"
28	有効	"show port table all"
29	有効	"show port info"
30	有効	"show port utilization table"
31	有効	"show data-path congestion"
32	無効	"show npu details"
33	無効	"show lagmgr details"
34	有効	"show fans"
36	有効	"show power chassis"
37	有効	"show temperature"
39	無効	"show alarm audible"
40	無効	"show alarm central-office"
41	無効	"show alarm outstanding"
42	無効	"show alarm statistics"
43	有効	"show cpu table"
44	無効	"show cpu info verbose"
45	有効	"show cpu errors verbose"
46	有効	"show cpu performance verbose"
47	無効	"show resources"
48	無効	"show task table"
49	無効	"show task memory"
50	無効	"show task memory max"
51	無効	"show task resources"
52	無効	"show task resources max"
53	有効	"show crash list"
54	有効	"show crash all"

番号	デフォルト SDR	コマンド文字列
55	無効	"show persistdump list"
72	無効	"show persistdump display"
57	有効	"show snmp trap history verbose"
58	無効	"show snmp trap statistics verbose"
59	有効	"show logs"
63	無効	"show messenger settings"
64	有効	"show messenger nameservice"
65	有効	"show messenger statistics"
66	有効	"show messenger bounces"
67	無効	"debug limits checkup detailed"
68	無効	"show plugin"
69	無効	"show module"
70	無効	"show ppp statistics"
71	無効	"show rsvp statistics"
72	有効	"show session disconnect-reasons verbose"
73	無効	"show apn statistics all"
74	無効	"show ipsg statistics"
75	無効	"show pdsn-service all"
76	無効	"show hsgw-service all"
77	無効	"show hsgw-service statistics all"
78	無効	"show epdg-service all counters"
79	無効	"show epdg-service statistics"
80	無効	"show fa-service all"
81	無効	"show ha-service all"
82	無効	"show mag-service all"
83	無効	"show mipv6ha-service all"
84	無効	"show lma-service all"
85	無効	"show dhcp-service all"
86	無効	"show sgsn-service all"

番号	デフォルト SDR	コマンド文字列
87	無効	"show sgsn sessmgr all memory statistics"
88	無効	"show operator-policy all"
89	無効	"show call-control-profile all"
90	無効	"show apn-profile all"
91	無効	"show imei-profile all"
92	無効	"show gprs-service all"
93	無効	"show iups-service all"
94	無効	"show sgtp-service all"
95	無効	"show map-service all"
96	無効	"show gs-service all"
97	無効	"show ggsn-service all"
98	無効	"show ggsn-service sgsn-table"
99	無効	"show cscf service all"
100	無効	"show cscf service diameter policy-control statistics"
101	無効	"show cscf service diameter location-info statistics"
102	無効	"show cscf service li-packet-cable statistics"
103	無効	"show cscf peer-servers full"
104	無効	"show demux-mgr statistics cscfmgr all"
105	無効	"show lac-service all"
106	無効	"show lns-service all"
107	無効	"show pdnclosedrp-service all"
108	有効	"show subscriber summary"
109	有効	"show connproxy sockets all"
110	無効	"show session progress"
111	無効	"show session subsystem data-info verbose"
112	無効	"show session subsystem full data-info"
113	無効	"show session subsystem facility sessmgr all debug-info"
114	無効	"show sessctrl config-reconciliation statistics"
115	無効	"show rp statistics"

番号	デフォルト SDR	コマンド文字列
116	無効	"show mipfa statistics"
117	無効	"show mipha statistics"
118	無効	"show mipv6ha statistics"
119	無効	"show lma-service statistics"
120	無効	"show mag-service statistics"
121	無効	"show cli configuration-monitor"
122	有効	"show srp info"
123	有効	"show srp checkpoint statistics"
124	無効	"show srp checkpoint statistics verbose"
125	無効	"show srp checkpoint statistics sessmgr all"
126	無効	"show srp checkpoint statistics ipsecmgr all"
127	有効	"show srp checkpoint statistics sessmgr all write-list-stats"
128	無効	"show srp monitor"
129	有効	"show srp monitor all"
130	無効	"show srp monitor diameter debug"
131	有効	"show srp statistics"
132	無効	"show srp call-loss statistics"
133	無効	"show srp audit-statistics"
134	無効	"show gtpc statistics verbose"
135	有効	"show gtpu statistics verbose"
136	有効	"show gtpu debug-info"
137	有効	"show gmm-sm statistics verbose"
138	有効	"show sgtpc statistics verbose"
139	有効	"show sgtpu statistics"
140	無効	"show ss7-routing-domain all sctp asp all status peer-server all peer-server-process all verbose"
141	有効	"show ss7-routing-domain all sctp asp all statistics gen"
142	無効	"show ss7-routing-domain all m3ua status peer-server all"
143	無効	"show ss7-routing-domain all m3ua statistics peer-server all peer-server-process all"

番号	デフォルト SDR	コマンド文字列
144	無効	"show ss7-routing-domain all qsaal statistics linkset all link all"
145	無効	"show ss7-routing-domain all sscf statistics linkset all link all"
146	無効	"show ss7-routing-domain all mtp3 status linkset all link all"
147	無効	"show ss7-routing-domain all mtp3 statistics gen"
148	無効	"show ss7-routing-domain all mtp3 statistics linkset all link all"
149	無効	"show ss7-routing-domain all routes"
150	無効	"show sccp-network all status all"
151	無効	"show global-title-translation association"
152	無効	"show global-title-translation address-map"
153	有効	"show egtpc peers"
154	無効	"show egtpc statistics interface mme"
155	有効	"show egtpc statistics interface sgsn"
156	有効	"show egtpc statistics interface sgw-ingress"
157	有効	"show egtpc statistics interface sgw-egress"
158	有効	"show egtpc statistics interface pgw-ingress"
159	有効	"show egtpc statistics interface cgw-egress"
160	有効	"show egtpc statistics interface epdg-egress"
161	無効	"show egtp-service all"
162	無効	"show gtpu-service all"
163	無効	"show pgw-service all"
164	無効	"show sgw-service all"
165	無効	"show saegw-service all"
166	無効	"show hcnbgw-access-service statistics"
167	無効	"show hcnbgw-network-service statistics"
168	無効	"show mme-service all"
169	無効	"show mme-service enodeb-association full all"
170	無効	"show mme-service statistics debug"
171	無効	"show mme-service db statistics"
172	無効	"show sgs-service all"

番号	デフォルト SDR	コマンド文字列
173	無効	"show sgs-service vlr-status full"
174	無効	"show sgs-service statistics all"
175	有効	"show sgw-service statistics all"
176	無効	"show saegw-service statistics all verbose"
177	無効	"show saegw-service statistics all function sgw verbose"
178	無効	"show saegw-service statistics all function pgw verbose"
179	有効	"show pgw-service statistics all"
180	無効	"show sccp statistics"
181	無効	"show tcap statistics"
182	無効	"show map statistics"
183	無効	"show sms statistics"
184	無効	"show pdg-service statistics"
185	無効	"show hnbgw sessmgr all memory statistics"
186	無効	"show hnbgw sessmgr all internal statistics"
187	無効	"show hnbgw disconnect-reasons"
188	無効	"show cs-network statistics"
189	無効	"show ps-network statistics"
190	無効	"show hnbgw statistics"
191	無効	"show hnbgw counters"
192	無効	"show demux-mgr statistics hnbmgr full"
193	無効	"show demuxmgr statistics bngmgr all"
194	無効	"show alcap statistics"
195	無効	"show pdg-service statistics micro-tunnel"
196	無効	"show pdg-service statistics transport"
197	無効	"show demuxmgr statistics al1mgr all"
198	無効	"show demuxmgr statistics famgr all"
199	無効	"show demuxmgr statistics hamgr all"
200	無効	"show demuxmgr statistics l2tpmgr all"
201	無効	"show demuxmgr statistics ipsgmgr all"

番号	デフォルト SDR	コマンド文字列
202	有効	"show demuxmgr statistics sgtpcmgr all"
203	無効	"show demuxmgr statistics imsimgr all"
204	有効	"show demuxmgr statistics gtpcmgr all"
205	有効	"show demuxmgr statistics egtpinmgr all"
206	無効	"show demuxmgr statistics egtpegmgr all"
207	無効	"show demuxmgr statistics pdgdmgr all"
208	有効	"show demuxmgr statistics gtpumgr all"
209	無効	"show bcmcs statistics all"
210	有効	"show linkmgr all parser statistics all"
211	無効	"show gtp accounting servers"
212	無効	"show gtp statistics verbose"
213	無効	"show gtp counters all"
214	無効	"show gtp storage-server"
215	無効	"show gtp storage-server statistics verbose"
216	無効	"show gtp storage-server local file statistics verbose"  <b>重要</b> このコマンドを設定した後で、設定を保存してからシャーンをリロードし、コマンドを有効にする必要があります。設定ファイルを保存してシャーンをリロードする方法については、使用している展開の『 <i>System Administration Guide</i> 』を参照してください。
217	無効	"show gtp storage-server local file counters all"  <b>重要</b> このコマンドを設定した後で、設定を保存してからシャーンをリロードし、コマンドを有効にする必要があります。設定ファイルを保存してシャーンをリロードする方法については、使用している展開の『 <i>System Administration Guide</i> 』を参照してください。
218	無効	"show gtp storage-server streaming file statistics verbose"
219	無効	"show gtp storage-server streaming file counters all"
220	無効	"show gtp group all"
221	有効	"show hd-storage-policy statistics all verbose"



番号	デフォルト SDR	コマンド文字列
222	有効	"show hd-storage-policy counters all verbose"
223	無効	"show dhcp statistics verbose"
224	無効	"show npu table"
225	無効	"show npu sf hw-info"
229	無効	"show l2tp statistics"
230	有効	"show fabric asr5500"
231	有効	"show vpn subsystem facility vpnmgr"
232	有効	"show session recovery status verbose"
233	有効	"show clock all"
234	無効	"show snmp statistics verbose"
235	無効	"show llc statistics verbose"
236	無効	"show bssgp statistics verbose"
237	無効	"show bssap+ statistics verbose"
238	無効	"show network-service-entity ip-config"
239	無効	"show network-service-entity fr-config"
240	無効	"show gprsns statistics sns-msg-stats"
241	無効	"show radius authentication servers detail"
242	無効	"show radius accounting servers detail"
243	有効	"show radius counters all"
245	有効	"show rct stats"
246	有効	"show heartbeat stats hatcpus"
247	無効	"show ntp associations all"
248	無効	"show npu details"
249	無効	"show active-charging service all"
250	無効	"show active-charging tcp-proxy statistics all verbose debug-info"
251	無効	"show active-charging edr-udr-file flow-control-counters verbose debug-only"
252	無効	"show active-charging service statistics"
253	無効	"show active-charging analyzer statistics"

番号	デフォルト SDR	コマンド文字列
254	無効	"show active-charging dns-learnt-ip-addresses statistics sessmgr all verbose"
255	無効	"show active-charging analyzer statistics name ip verbose"
256	無効	"show active-charging analyzer statistics name ipv6 verbose"
257	無効	"show active-charging analyzer statistics name tcp verbose"
258	無効	"show active-charging analyzer statistics name http verbose"
259	無効	"show active-charging charging-action statistics"
260	無効	"show active-charging rulebase statistics"
261	無効	"show active-charging ruledef statistics all charging"
262	有効	"show active-charging ruledef statistics all firewall wide"
263	無効	"show active-charging regex status all"
264	無効	"show active-charging regex statistics memory summary"
265	無効	"show active-charging regex statistics ruledef summary"
266	無効	"show active-chargingedr-format statistics"
267	無効	"show active-charging subsystem all debug-only"
268	無効	"debug acsmgr show flow-stats cumulative all-flows"
269	無効	"debug acsmgr show flow-stats cumulative http"
270	無効	"debug acsmgr show flow-stats cumulative ip"
271	無効	"debug acsmgr show flow-stats cumulative tcp"
272	無効	"debug acsmgr show flow-stats cumulative udp"
273	無効	"debug acsmgr show flow-stats max-simultaneous-flows all-flows"
274	無効	"debug acsmgr show flow-stats max-simultaneous-flows http"
275	無効	"debug acsmgr show flow-stats max-simultaneous-flows ip"
276	無効	"debug acsmgr show flow-stats max-simultaneous-flows tcp"
277	無効	"debug acsmgr show flow-stats max-simultaneous-flows udp"
278	無効	"debug acsmgr show flow-stats duration-based all-flows"
279	無効	"debug acsmgr show flow-stats duration-based tcp"
280	無効	"debug acsmgr show flow-stats duration-based udp"
281	無効	"debug acsmgr show flow-stats lifetime-based all-flows"
282	無効	"debug acsmgr show p2p detection-params sct"

番号	デフォルト SDR	コマンド文字列
283	無効	"debug acsmgr show rule-optimization-information"
284	無効	"debug sessmgr charging-service show-stats all"
285	無効	"debug acsmgr show memory usage"
286	無効	"debug aaamgr show memory usage"
287	無効	"show active-charging credit-control statistics debug-info"
288	無効	"show active-charging credit-control session-states"
289	無効	"show active-charging credit-control statistics"
290	無効	"show diameter endpoints all"
291	無効	"show diameter endpoints all debug-info"
292	無効	"show diameter route table debug-info"
293	無効	"show diameter peers full debug"
294	無効	"show diameter aaa-statistics"
295	無効	"show diameter aaa-statistics all"
296	無効	"show diameter aaa-statistics debug-info"
297	無効	"show diameter accounting servers debug-info"
298	無効	"show diameter authentication servers debug-info"
299	無効	"show diameter statistics"
300	無効	"show diameter statistics debug-info"
301	無効	"show diameter statistics proxy"
302	無効	"show diameter statistics proxy debug-info"
303	無効	"show diameter dynamic-dictionary all contents"
304	無効	"show active-charging edr-udr-file statistics"
305	無効	"show active-charging firewall statistics debug-info"
306	無効	"show active-charging nat statistics"
307	無効	"show demuxmgr statistics asngwmgr all"
308	無効	"show asngw-service all"
309	無効	"show asngw-service statistics verbose"
310	無効	"show demuxmgr statistics asnpcmgr all"
311	無効	"show asnpc-service all"

番号	デフォルト SDR	コマンド文字列
312	無効	"show asnpc-service statistics verbose"
313	無効	"show demuxmgr statistics phsgwmgr all"
314	無効	"show phsgw-service all"
315	無効	"show phsgw-service statistics verbose"
316	無効	"show demuxmgr statistics phspcmgr all"
317	無効	"show phspc-service all"
318	無効	"show phspc-service statistics verbose"
319	無効	"show demuxmgr statistics magmgr all"
320	無効	"show active-charging content-filtering category policy-id all"
321	無効	"show content-filtering category database all verbose"
322	無効	"show content-filtering category database facility srdmgrp all verbose"
323	無効	"show content-filtering category statistics"
324	無効	"show content-filtering category statistics facility srdmgrp all"
325	無効	"show active-charging content-filtering category statistics"
326	無効	"show active-charging content-filtering server-group statistics verbose"
327	無効	"show active-charging url-blacklisting statistics"
328	無効	"show url-blacklisting database all"
329	無効	"show url-blacklisting database facility acsmgr all"
330	無効	"show active-charging tethering-detection database"
331	無効	"show active-charging tethering-detection database sessmgr all"
332	無効	"show active-charging tethering-detection statistics"
333	無効	"show ims-authorization service statistics"
334	無効	"show ims-authorization policy-control statistics"
335	無効	"show ims-authorization policy-control statistics debug-info"
336	無効	"show local-policy statistics summary"
337	無効	"show rohc statistics"
338	無効	"show dns client statistics"
339	無効	"show hss-peer-service service all"
340	無効	"show ipms status all"

番号	デフォルト SDR	コマンド文字列
341	無効	"show ipms status debug-info"
342	無効	"show kvstore"
343	無効	"show kvstore verbose"
344	無効	"show kvstore kvclient"
345	無効	"show kvstore kvmgr"
346	無効	"show pcc-service all"
347	無効	"show pcc-service statistics all"
348	無効	"show pcc-policy service all"
349	無効	"show pcc-policy service statistics all"
350	無効	"show pcc-quota service all"
351	無効	"show pcc-quota service statistics all"
352	無効	"show pcc-af service all"
353	無効	"show pcc-af service statistics all"
354	無効	"show pcc-sp-endpoint all"
355	無効	"show pcc-sp-endpoint statistics all"
356	無効	"show event-notif-list server all"
357	無効	「show event-notif-list statistics」
358	無効	"show demux-mgr statistics bindmux all"
359	無効	"show congestion-control configuration"
360	無効	"show congestion-control statistics mme full"
361	無効	"show congestion-control statistics imsimgr all full"
365	無効	"show cli history"

注：

- 有効 = デフォルトレコードセクションに含まれています
- 無効 = デフォルトレコードセクションに含まれていません



## 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。