# VPC-DI System リリース 21.21 アドミニストレーション ガイド

**初版**：2020 年 10 月 1 日

**最終更新**：2020 年 10 月 15 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/**）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。

# 目 次

# このマニュアルについて

ここでは、『*VPC-DI System 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 仮想パケットコアの分散インスタンス（VPC-DI）は、複数の仮想マシン（VM）の完全に分散されたネットワークで構成され、管理、入出力（I/O）、およびパケット処理を実行する VM を持つ単一の StarOS インスタンスを形成します。VM は、既製品（COTS）のサーバ上で動作します。このガイドでは、VPC のさまざまなコンポーネントを設定および管理する方法について説明します。

（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す。

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 （xxxiii ページ）
- 関連資料 （xxxv ページ）
- カスタマー サポートへの問い合わせ （xxxvi 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。 |

| 通知タイプ | 説明 |
|---|---|
| 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、電気的障害のおそれがあることを警告する場合もあります。 |

| 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。<br>ログイン： |
| として表現されるテキスト **commands** | 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。<br>**show ip access-list**<br>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。 |
| **command** 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。<br>**show card** *slot_number*<br>*slot_number* 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。<br>[File] メニュー、[New] の順にクリックてください。 |

| コマンドシンタックスの表記法 | 説明 |
| --- | --- |
| **{ keyword** 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。<br>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。<br>**sctp-max-data-chunks { limit** *max_chunks* **\| mtu-limit }**<br>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。<br>**snmp trap link-status** |
| **[ keyword** 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。<br>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。<br>**action activate-flow-detection { intitiation \| termination }**<br>または<br>**ip address [ count** *number_of_packets* **\| size** *number_of_bytes* **]** |

# 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 では、次のユーザドキュメントを入手できます。

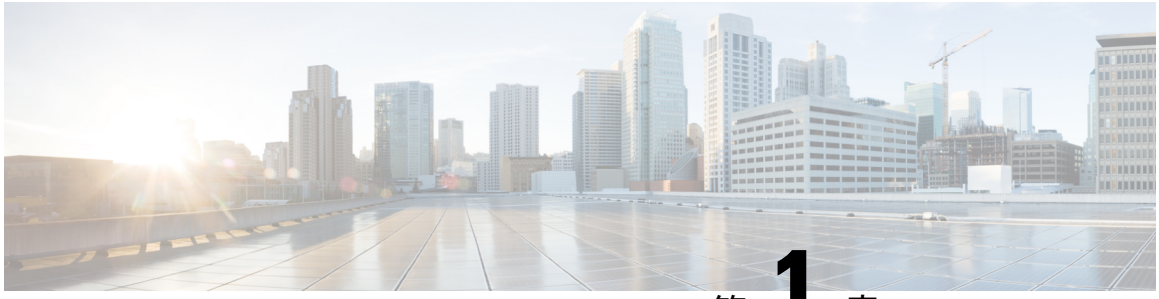
- *AAA Interface Administration and Reference*
- *Command Line Interface Reference*
- *GTPP Interface Administration and Reference*
- *IPSec Reference*
- *Release Change Reference*

- *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-DI の概要

この章では、Cisco Virtualized Packet Core—Distributed Instance（VPC-DI）について説明します。VPC-DI は、単一の仮想マシン（VM）を超えて境界を拡張することにより、仮想化されたクラウドアーキテクチャのスケーラビリティをサポートします。



## 製品の説明

この章では、StarOS VPC-DI 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-DI は、1 つの仮想マシン（VM）という枠を超えて仮想化された StarOS を配布します。これにより、共有インターフェイス、共有サービスアドレス、ロードバランシング、冗長性、および管理の一元化を使用して、複数の VM を 1 つの StarOS インスタンスとして機能させることができます。

システムは、1 つの StarOS インスタンスを形成するためにグループ化された複数の VM から成る完全分散ネットワークとして動作し、各タイプのスタンバイ VM で管理およびセッション処理を実行する VM を備えています。

# システムの基礎となるインフラストラクチャ

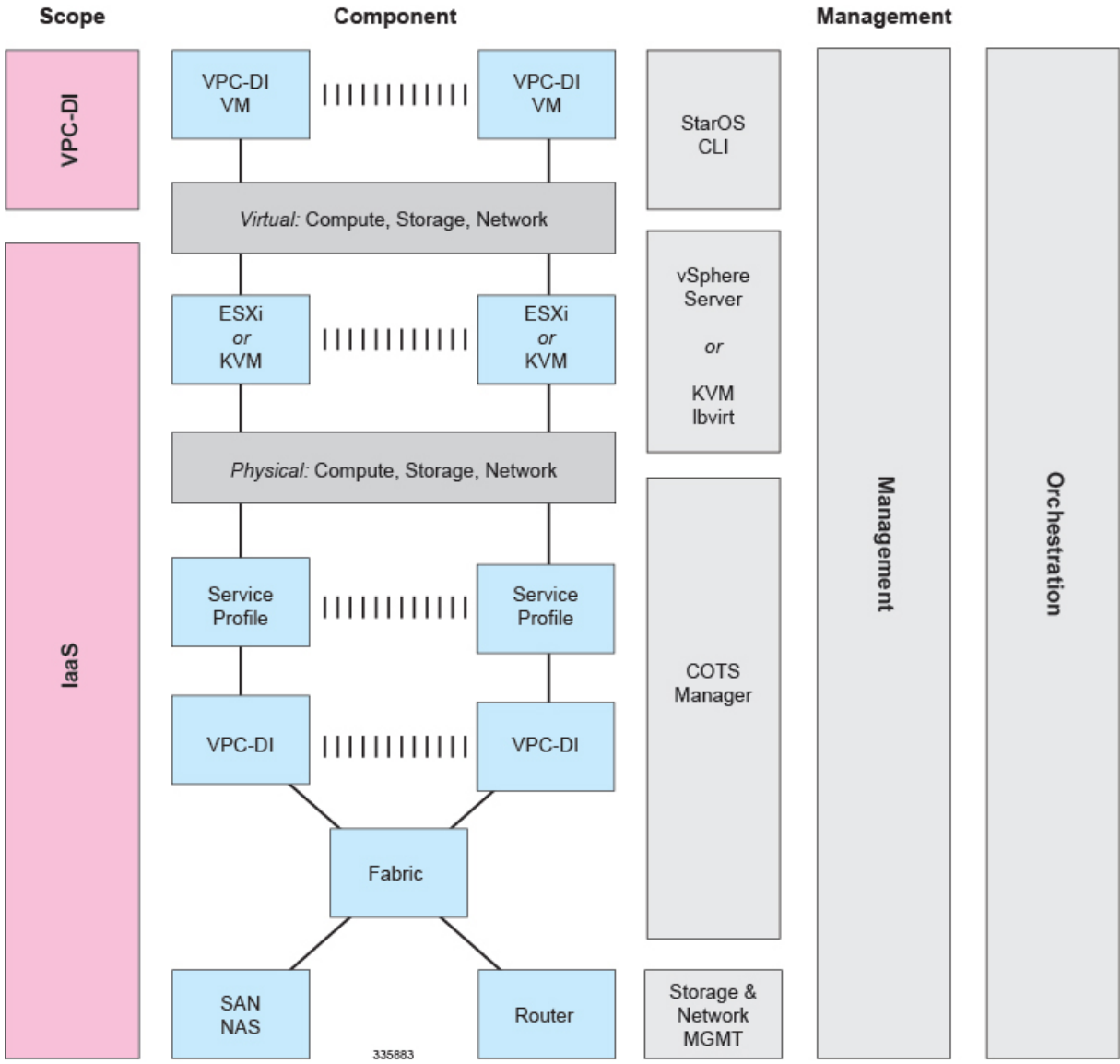
この仮想化システムは、新規または既存の Infrastructure as a Service（IaaS; サービスとしてのインフラストラクチャ）のクラウドデータセンターに展開できます。VPC-DIは、既製品（COTS）のサーバで業界標準のハイパーバイザを使用して、一連の仮想マシン（VM）で実行されます。この導入モデルでは、物理インフラストラクチャの管理をStarOSとVPC-DIの範囲外に維持することができ ます。

一般的なインスタンスは、次の NFVi（ネットワーク機能の仮想インフラストラクチャ）で実行されます。

- IaaS コンポーネント
  - COTS ブレードシャーシ、ブレード、およびファブリックインターコネクト
  - Manager ソフトウェア
  - ネットワーク接続ストレージ（NAS）またはストレージエリアネットワーク（SAN）
  - 各ブレードまたはサーバ上の VMWare ESXi または KVM ハイパーバイザ
  - VMware vSphere サーバまたは KVM OpenStack
- VM 内にインストールされた StarOS ソフトウェア
- ブレードに障害が発生した場合、単一のVMのみが影響を受けるように、各VMは別々のブレード上で実行する必要があります（アンチアフィニティ）。
- 既存の管理ソフトウェア（サービス、ロギング、統計情報など）
- オーケストレーション ソフトウェア（オプション）（オーケストレーション （18 ページ） を参照）

VPC-DI インスタンスは、StarOS の単一の管理可能なインスタンスとして機能する VM のグループです。VPC-DI は、 次の主要なコンポーネントで構成されています。

図 *1 : VPC-DI* の範囲





- 制御機能（CF）の VM （3 ページ）
- サービス機能（SF） VM （4 ページ）
- DI ネットワーク （6 ページ）

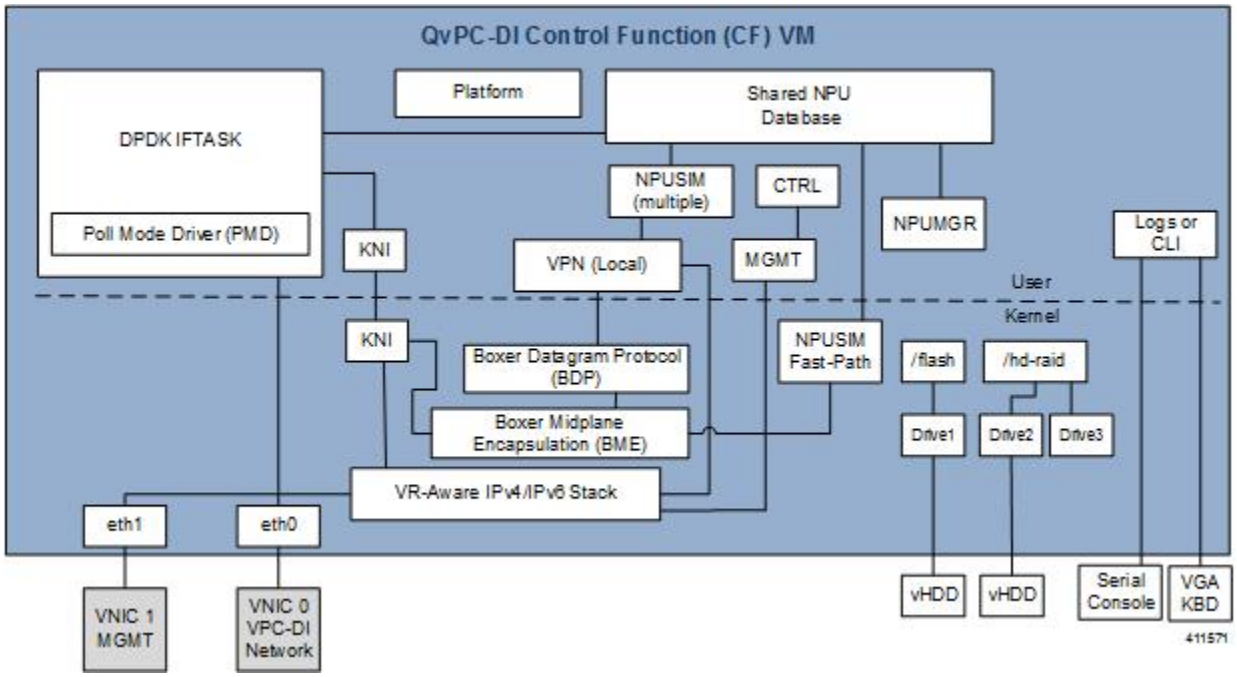

# 制御機能（CF）の VM

2 つの CF VM がアクティブ/スタンバイ（1:1）冗長ペアとして機能します。アクティブ CF は次の機能を果たします。

- コントローラのタスク

- ローカルコンテキスト VPNMGR
- ローカルコンテキスト（MGMT）と DI ネットワーク 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
- CLI およびロギング用のアウトオブバンド（OOB） 管理（vSerial と vKVM）

**図 *2 :* 制御機能 *VM***




# サービス機能（SF） VM

SF VM は、サービスコンテキスト（ユーザ I/O ポート）を提供し、プロトコルシグナリングおよびセッション処理タスクに対応します。VPC-DI インスタンスは最小 4、最大 14 の SF VM を持つことができ、最大 12 の SF VM をアクティブにできます。

各 SF VM は、CF に指定されているとおりに 3 つのロールのいずれかを動的に実行します。

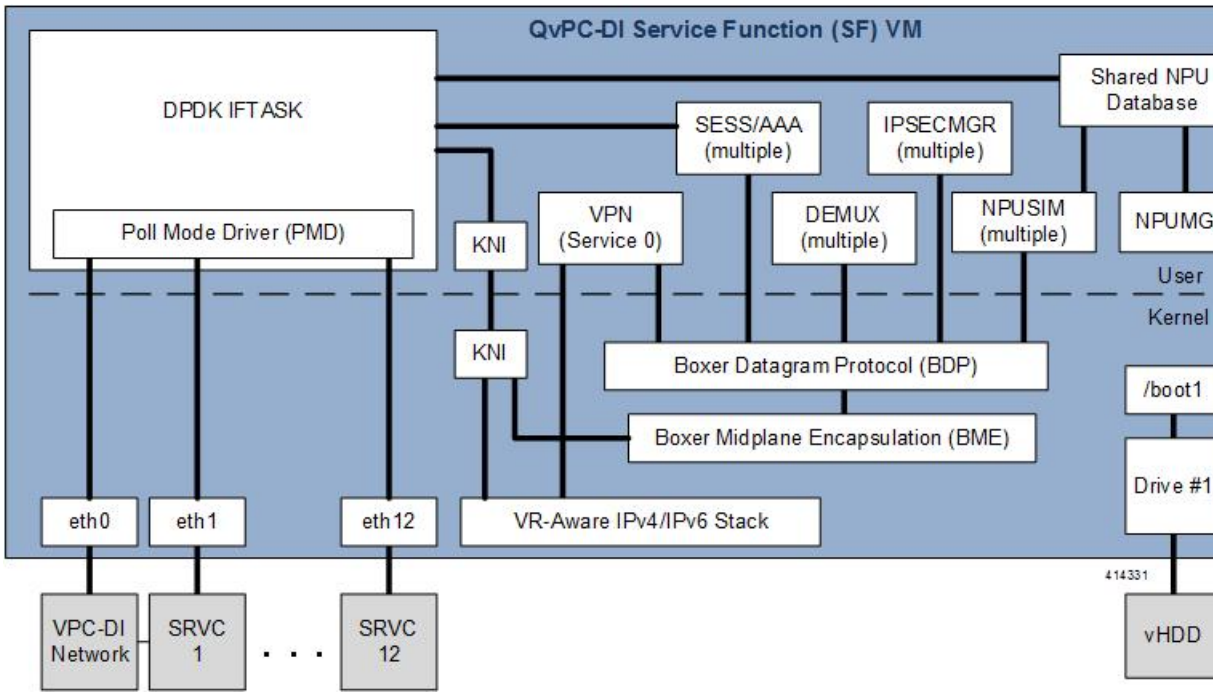
- Demux VM（フロー割り当て）
- セッション VM（トラフィック処理）
- スタンバイ VM（n+1 冗長性）

SF は次の機能を担当します。

| 機能 | 実行場所 |
|---|---|
| NPUSIM fastpath/slow パス（NPU エミュレーションと CPU へのルーティング） | Demux VM、セッション VM、スタンバイ VM |
| Intel® Data Plane Development Kit（DPDK）に基づく IFTASK | Demux VM、セッション VM、スタンバイ VM |
| 非ローカルコンテキスト（SRVC）の vNIC ポート | Demux VM、セッション VM、スタンバイ VM |
| サービスコンテキスト（最初の VM）の VPNMGR および Demux | Demux VM |
| セッション処理のための SESSMGR および AAAMGR（追加の VM） | セッション VM |
| 出力転送の決定 | |
| 暗号処理 | |

VPC-DI インスタンスの最小設定では、4 つの SF の 2 つのアクティブ、1 つの demux 、および 1 つのスタンバイが必要です。

**図 *3 :* サービス機能 *VM***



# DI ネットワーク

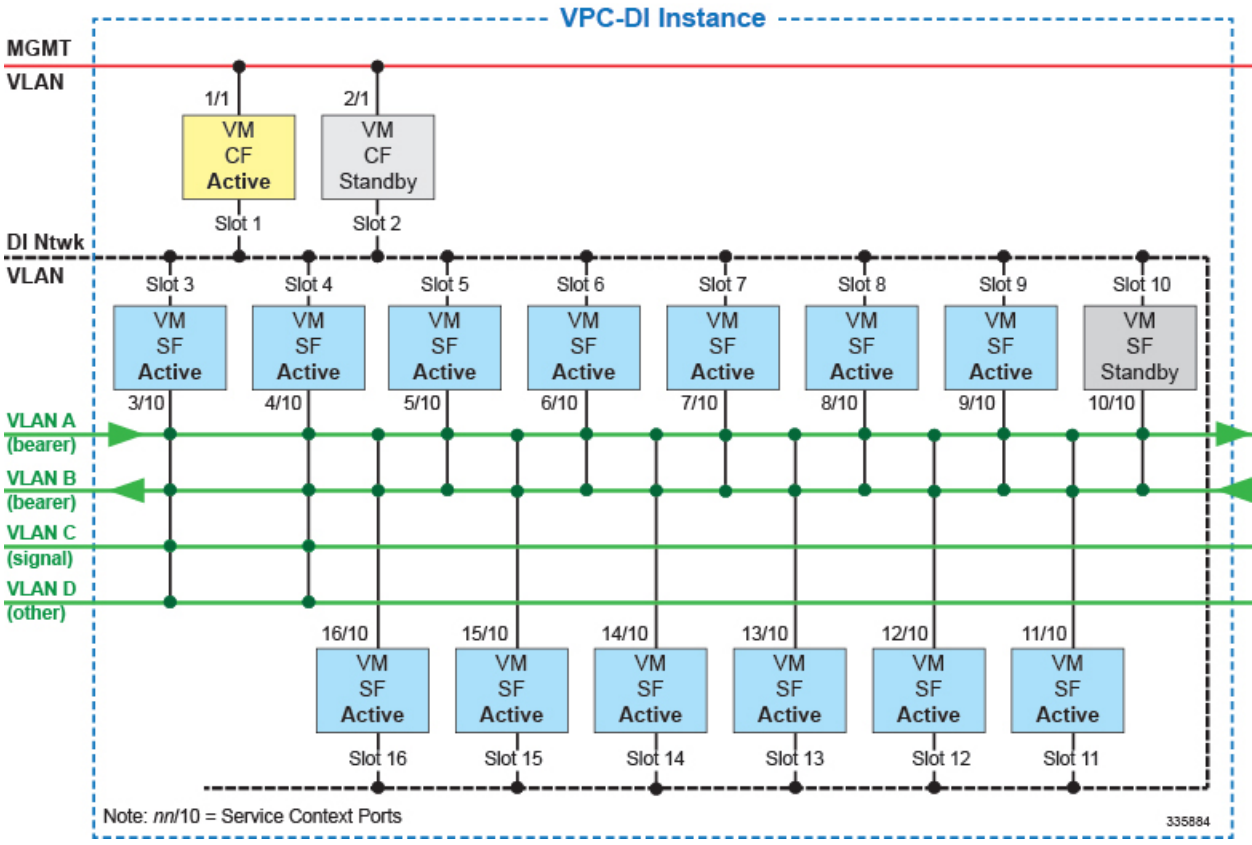
VPC-DI インスタンス内の VM が相互に通信するためには、各インスタンスに VM をインターコネクトするプライベート L2 ネットワークが必要です。このネットワークは、IaaS/仮想化インフラストラクチャ内の VLAN を使用し、最初の vNIC として各 VM にタグなしで公開されるようにする必要があります。

DI ネットワークは、単一の VPC-DI インスタンスを排他的に使用する必要があります。他のデバイスがこのネットワークに接続されていない可能性があります。アクセスを制限するセキュアなネットワークである必要があります。

（注） 1 つのデータセンター内で複数のインスタンスをインスタンス化する場合は、各インスタンスに独自の DI ネットワークが必要です。

インスタンス内のすべての VM は、物理的に同じサイトに配置する必要があり、理想的には最小限の相互接続デバイスとともに同じいくつかのラックに配置します。DI ネットワークの信頼性は、VPC-DI インスタンスの安定性を確保するために重要です。WAN または輻輳状態のリンク全体で L2 トンネリングプロトコルを使用することは推奨されません。

図 *4 : DI* ネットワーク



# ネットワーク要件

DI ネットワークの信頼性とパフォーマンスは、VPC-DI の信頼性とパフォーマンスにとって重要です。DI ネットワークは、内部制御、シグナリング、およびベアラートラフィックに使用されます。ベアラートラフィックは DI ネットワークを複数回通過する可能性があるため、DI ネットワーク内のパケット損失が認識可能な VPC-DI のパケット損失全体に影響します。

OpenStack での展開の場合、DI ネットワークとサービスネットワークの両方のタイプを SR-IOV モードフラットとして設定する必要があります。VMware での展開の場合、DI ネットワークタイプは、ボンディングおよび VLAN タグ付きの VMXNET3 または PCI-PT として設定する必要があります。VMware では、サービスネットワークタイプを VMXNET3 または PCI-PT として設定する必要があります。

（注） VM を接続するインフラストラクチャは、すべての VM 間で 10 Gbps 以上であり、冗長性の設定が行われている必要があります。冗長性の設定は、次のいずれかの方法で実装できます。

- vSwitch を使用しているホスト上（Virtio/VMXNET3 インターフェイスの場合）
- Cisco UCS 仮想インターフェイスカード（VIC）などのハードウェア上
- ネットワーク インターフェイスのボンディンを使用している VPC-DI 内

IaaS/ハイパーバイザには、次が可能な DI ネットワークが備わっている必要があります。

- L2 イーサネットブリッジ/スイッチとして実行できる。
- 少なくとも 7100 バイトのジャンボフレームをサポートできる。インストールでジャンボフレームがサポートされていない場合でも、VPC-DI を使用できます。サポート対象 MTU を超えるトラフィックのサポートの設定 （39 ページ）の説明に従って、起動パラメータファイルに適切なパラメータを設定する必要があります 。

インフラストラクチャ/ハイパーバイザには、次が可能な DI ネットワークが備わっている必要があります。

- VID = 0 の VM から送信された優先順位 802.1p をサポートしている。
- インスタンス内のすべての VM 間で優先順位 802.1p のエンドツーエンドで受け入れる。
- すべての物理インフラストラクチャ内に冗長 L2 パスが備わっているか、またはインスタンス内のすべてのシステム VM 間で優先順位 802.1p のエンドツーエンドが備わっている。
- アクセスを制限するセキュアなネットワークが備わっている。ローカルでのみ使用できるという意味で、セキュアなネットワークである必要があります。

具体的には、DI ネットワークには次の最低限の信頼性要件が必要です。

- 完全冗長 L2 パス
- STP や LACP の停止を含めて、1.5 秒を超える停止がない（該当する場合）

- パケットの優先順位付け
- 制御パケットまたはベアラーパケットの損失を最小限に抑えるために十分なネットワーク帯域幅

DI ネットワーク内での中断や過度なパケット損失が原因で、VPC-DI インスタンスでの障害の誤検出や、想定外の動作が発生する場合があります。

各システム VM は、他の VM の到達可能性や DI ネットワークの信頼性を継続的にモニタします。

## ジャンボ フレーム

DI ネットワークでは、少なくとも 7100 バイトのジャンボフレームをサポートすることを推奨します。起動時に、各 VM はネットワーク上で一連の ping コマンドを発行して、ジャンボフレームがサポートされているかどうかを確認します。ジャンボフレームがサポートされていれば、システムパフォーマンスが向上します。

インストールでジャンボフレームがサポートされていない場合でも、VPC-DI を使用できます。サポート対象 MTU を超えるトラフィックのサポートの設定 （39 ページ）の説明に従って、起動パラメータファイルに適切なパラメータを設定する必要があります。

7100 未満の MTU が検出され、適切な起動パラメータが設定されていない場合、CF と SF は起動しません。

また、StarOS 設定で適切に設定されている場合、SF のサービスポートは、リリース 21.4 以降で最大 9,100バイトの最大 MTU、古いリリースでは 2,048 バイトをサポートできます。

## 記録ストレージ

記録ストレージは、**/records** で使用可能なインスタンス全体のストレージデバイスで使用できます。両方の CF VM には、記録ストレージに適したサイズ（最小 16 GB）の 2 番目の vHDD（/hd-raid）がプロビジョニングされます。CF は RAID 設定を共有して、vHDD 間のデータをミラーリングします。SF は、DI ネットワークを介してアクティブ CF にデータレコードを送信し、手動で作成しマウント、または VNFM によってオーケストレーションした外部の一時的なストレージに転送します。

# パケットフロー

SF ポートを使用して、ベアラーとシグナリングパケットを送受信します。ネットワーク設定とアドレスの使用をシンプルにするため、すべての SF に接続する必要があるのは、高帯域幅（ベアラー）パケット用の VLAN のみです。低帯域幅インターフェイス（シグナリング）は 2 つの SF のみに接続できます。次の図では、ベアラー VLAN はすべての SF に接続され、シグナリングとその他の VLAN は最初の 2 つの SF のみに接続されています。

（注） この非対称の配置は、必要なインターフェイスの数は少なくなりますが、2 つの VM の損失によりサービスが損失することになるため、障害については慎重に検討する必要があります。

ECMP はハッシュに基づく負荷分散を行い、任意の SF VM にトラフィックを送信できます。

入力では、SF はフロールックアップを実行し、特定の SF 上の特定の SESSMGR タスクにパケットを転送します。この入力トラフィックの一部は、ローカルSESSMGRタスクによって処理されます。それ以外の場合は、DI ネットワークを介して正しい SF にリレーされます。出力では、各 SF がローカルポートからパケットを送信します（ECMP を使用している場合）。ほとんどの場合、パケットが通過する VM の数は 2 未満です。ただし、ACL とトンネリングにより、EPC の設定に応じて、特定のフローのホップ数を増加させる可能性があります。

## SF Demux VM で受信したパケット

demux とスタンバイ SF では、受信したすべてのセッショントラフィックが別の SF にリレーされ、セッション処理が行われます。次の図は、入力パケットが Demux SF を介して他のセッション SF にどのように分散されて処理されるかを示しています。

| 項目 | 説明 |
|---|---|
| 1 | 受信 NPU はフローのルックアップを行い、SESSMGR を特定します（入力）。 |
| 2 | SESSMGR はパケットを処理します。 |
| 3 | 送信 NPU はローカルポートにパケットを送信します。（出力） |

## SF セッション VM で受信したパケット

次の図に、セッション SF が受信した入力パケットをどのように他のセッション SF に配信して処理するかを示します。

| 項目 | 説明 |
|---|---|
| 1 | 受信 NPU はフローのルックアップを行い、SESSMGR を特定します（入力）。 |
| 2 | SESSMGR はパケットを処理します。 |
| 3 | 送信 NPU はローカルポートにパケットを送信します。（出力） |

# 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

VPC-DI インスタンス内の各 VM は、1 つの 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。

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、ASR 5500 プラットフォームの場合と同様に各 VM 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。

vNIC の順序を手動で指定するには、起動パラメータファイルの作成 （23 ページ）を参照してください。

デフォルトでは、システムはハイパーバイザが提示する順序で vNIC インターフェイスを割り当てます。

- CF VM（スロット 1 および 2）
  - 最初に提示されるインターフェイス（1/0 または 2/0）は DI ネットワーク用です。
  - 2 番目に提示されるインターフェイス（1/1 または 2/1）は管理ネットワーク用です。
- SF VM（スロット 3 ～ 16）
  - 最初に提示されるインターフェイス（*slot*/0）は DI ネットワーク用です。
  - トラフィックインターフェイス *slot*/10 ～ *slot*/21 は、IaaS VLAN 制御トラフィックとデータトラフィック用です。

（注） StarOS は最大 12 個のサービスポートをサポートしますが、実際のポート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。

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と、KVM ブリッジグループまたは VMware vSwitch が VM インターフェイスと一致させる順序で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 VM 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 vNIC 設定に表示されている MAC アドレスを調べ、VM 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、VM インターフェイスが目的のブリッジグループまたは VMware vSwitch に接続されていることが確認されます。

# 暗号化

VPC-DI 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。

COTS サーバハードウェアが Intel 89xx チップに基づいて Coleto Creek のチップセットを使用している場合、システムはパケットの暗号化と復号化にこのハードウェアチップを自動的に利用します。ただし、システムが暗号化と復号化にハードウェアチップセットを使用するためには、すべてのサービス機能 Vm でこのチップセットを使用する必要があります。

## 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-DI 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。

シャーシ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。

# CF 冗長性

2 つの CF VM は、VPC-DI インスタンスおよびローカルコンテキストポートやローカル管理ポートを制御するために 1:1 冗長化されています。

両方の CF の管理ポート vNIC は相互に 1:1 冗長であり、インフラストラクチャ内の同じ VLAN に配置する必要があります。一度に 1 つの管理ポートのみがアクティブになります。

（注） ホストまたはハイパーバイザに障害が発生した場合に冗長性を確保するために、2 つの CF VM を同じ物理ホスト（サーバまたはブレード）上で実行することはできません。

# SF 冗長性

SF VM ごとに、サービスポートのネットワーク接続が提供されます。各 SF には 1 つ以上のポートと関連付けられたインターフェイスがありますが、SF はペアリングされていないため 1:1 の冗長性を提供しません。

SF ポートの冗長性は、ECMP またはサポートされている別の L3 プロトコルを使用して確立する必要があります。

インスタンスに必要な合計スループットは、セッションリカバリが有効になっている N-2 SF を超えないようにする必要があります。これにより、1 つの SF が失敗しても、他の SF が負荷を引き継ぐことができます。サービス IP アドレスにループバックインターフェイスを使用することを強くお勧めします。

SF とピアルータ間のパス障害を検出するために BFD を使用することをお勧めします。これにより、障害が発生した場合に ECMP パスが除外されます。

セッションリカバリが有効になっている場合、1 つの VM が VPN/Demux になり、残りの VM はセッション処理 VM になります。スタンバイ SF は、他の SF に応じて冗長性を提供できます。

（注） 各 SF VM は、ホストまたはハイパーバイザに障害が発生した場合に冗長性を確保するために、別の物理ホスト上で実行する必要があります。

## ICSR のサポート

VPC-DI 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-DI 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。

# ハイパーバイザ要件

VPC-DIは、次のハイパーバイザで実行する資格があります。

- OpenStack ベースの仮想環境
- VMware ESXi
  - バージョン 6.0：リリース 21.8 より前のリリースでサポート
  - 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

Heat テンプレート（OpenStack 用）および OVF/OVA テンプレート（VMware vSphere 用）は、CF および SF VM 用に提供されています。

VMware vApp は、VPC-DI インスタンスの VM をまとめてバンドルし、簡単に展開できるようします。

OpenStack で VPC-DI を展開するには、Cisco Elastic Services Controller（ESC）を使用することをお勧めします。OpenStack での ESC を使用した VPC-DI のオンボーディング （40 ページ）を参照してください。

（注） 提供されるテンプレートからの価格偏差は、期待されるパフォーマンスと信頼性を維持するために、シスコのエンジニアリングによって承認される必要があります。

# CF VM 設定

システムでは、各 CF VM が次のように設定されている必要があります。

- 8 vCPU
- 16 GB RAM
- 最初の vNIC は DI ネットワークです。
- 2 番目の vNIC は管理ポートです。
- 最初の vHDD は、起動イメージと設定ストレージです（/flash、非 RAID、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）（hd-local1、RAID、最小 16 GB を推奨）

（注） 両方の CF VM が同一に設定されている必要があります。

# SF VM 設定

システムでは、各 SF VM が次のように設定されている必要があります。

- 12 個以上の vCPU（vCPU と vRAM のオプション （15 ページ） を参照）。

（注） SF VM は起動せず、vCPU のこの最小要件が満たされていない場合は、次のエラーが報告されます。

```
Found hardware disparate in minimum number of cores, found n
 cores, minimum expected cores are 12.
```

- 32 GB 以上の vRAM（vCPU と vRAM のオプション （15 ページ） を参照）。
- 最初の vNIC は DI ネットワークです。
- 2 番目とそれ以降の vNIC はサービスポートです。システムは最大 12 個の vNIC をサポートしますが、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。
- vHDD は起動イメージ用で、2 GB が推奨されています。

（注） すべての SF VM が同様に設定されている必要があります。VM ハードウェア設定のモニタリングの詳細については、VM ハードウェアの検証 （102 ページ）を参照してください。

# 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。

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 t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。すべての VM上で一度に再設定を実行できます。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：KMV 用の準仮想 NIC
- ixgbe：Intel 10 ギガビット NIC 仮想機能
- enic：Cisco UCS NIC
- SR-IOV：シングルルート I/O の仮想化 ixgbe インターフェイスと enic インターフェイス

# vhost-net および vhost-user のサポート

システムは、vhost-net と vhost-user ベースの両方のバックエンドメカニズムと対話できる DPDK ベースのユーザアプリケーションに基づいて Virtio フロントエンドを実装します。vhost-user と vhost-net は、共有メモリベースの、イベント、および割り込み記述子を使用する vhost パラダイムの実装を提供します。DPDK ベースのフロントエンドドライバを vhost-net と vhost-user と組み合わせて使用すると、Linux ブリッジベースのデータパスと比較して、パフォーマンスの高いデータパスを実現できます。

- vhost-user は、ユーザ空間でパケットを完全に処理し、パフォーマンスを引き上げます。システムは DPDK ベースのユーザ空間アプリケーションにフロントエンドを実装しますが、ホストユーザ空間アプリケーションは vhost-user インターフェイスに基づいてバックエンドを実装します。
- vhost-net は、Virtio ネットワーキングのカーネルレベルのバックエンドを提供します。これにより、Virtio パケット処理タスクをユーザ空間 (QEMU プロセス) から移動し、カーネル (vhost-net ドライバ) に移動することで、仮想化のオーバーヘッドを軽減します。これにより、デバイス エミュレーション コードは、ユーザ空間からシステムコールを実行する代わりに、カーネルサブシステムに直接コールすることができます。

  システムは、vhost-user で 1 つのキューをサポートします。

# 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**<br>（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 を使用します |

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

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 アーキテクチャの不可欠な部分であり、システム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、ユーザ空間でのパケット処理の高速パスを提供します。これはシステムの運用に必要です。CF または SF のインスタンス化時に、DPDK は CPU コアの合計数に応じて、CPU コアの特定の割合を IFTASK に割り当てます。残りの CPU コアはアプリケーションに割り当てられます。

IFTASK で使用されている CPU コアを特定し、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

Wednesday July 06 10:53:55 PDT 2017
          -------iftask-------
   lcore     now   5min  15min
--------  ------ ------ ------
  01/0/1     38%    53%    52%
  01/0/2     51%    55%    55%
  02/0/1     66%    72%    68%
  02/0/2     66%    63%    67%
  03/0/1     57%    55%    55%
  03/0/2     51%    47%    45%
  03/0/3     89%    89%    89%
  03/0/4     88%    88%    89%
  04/0/1     67%    59%    58%
  04/0/2     54%    40%    48%
  04/0/3     89%    89%    90%
  04/0/4     90%    89%    89%
  05/0/1     55%    55%    56%
  05/0/2     68%    45%    45%
  05/0/3     90%    89%    89%
  05/0/4     90%    89%    89%
  06/0/1     50%    58%    58%
  06/0/2     24%    24%    25%
  06/0/3     89%    90%    90%
  06/0/4     91%    90%    90%
```

IFTASK コアを使用せずに 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 6
Tuesday July 05 10:39:52 PDT 2017
Card 6, CPU 0:
  Status                : Active, Kernel Running, Tasks Running
  Load Average          : 7.74, 7.62, 7.54 (9.44 max)
  Total Memory          : 49152M
  Kernel Uptime         : 4D 5H 7M
  Last Reading:
    CPU Usage           : 25.4% user, 7.8% sys, 0.0% io, 0.1% irq, 66.7% idle
    Poll CPUs           : 4 (1, 2, 3, 4)
    Processes / Tasks   : 177 processes / 35 tasks
    Network             : 164.717 kpps rx, 1025.315 mbps rx, 164.541 kpps tx, 1002.149
mbps tx
    File Usage          : 8256 open files, 4941592 available
    Memory Usage        : 21116M 43.0% used
  Maximum/Minimum:
    CPU Usage           : 32.9% user, 8.9% sys, 0.0% io, 0.4% irq, 59.1% idle
    Poll CPUs           : 4 (1, 2, 3, 4)
    Processes / Tasks   : 184 processes / 36 tasks
    Network             : 178.388 kpps rx, 1270.977 mbps rx, 178.736 kpps tx, 1168.999
mbps tx
    File Usage          : 8576 open files, 4941272 available
    Memory Usage        : 21190M 43.1% used
```

# オーケストレーション

VPC-DI インスタンスが展開されると、VPC-DI が実行されている環境のいくつかの期待は、StarOS の制御を超えます。これらのほとんどは、オーケストレーションシステムの要件に分類されます。

- インストールとパラメータの割り当てなどの VPC-DI VM のプロビジョニング：各 VM の設定、接続、および永続ブロックストレージ。
- DI ネットワークが信頼性の要件を確実に満たすようにするための DI ネットワークの L2 プロビジョニング。
- ネットワーク分離のポリシー強制（該当する場合）。
- 冗長性ルールを強制する VM の物理的な配置。
- CPU、RAM、NIC などの物理リソースに便利なモニタリングツールを提供。

VPC-DI インスタンスの展開にオーケストレーションシステムを使用しない場合は、これらの要件を維持する必要があります。ただし、手動で、または他の方法で、強制的に適用する必要があります。VM ハードウェア設定のモニタリングの詳細については、VM ハードウェアの検証 （102 ページ）を参照してください。

# プロビジョニング

VPC-DI インスタンスのプロビジョニングには、次の 2 つのフェーズがあります。

- VM とネットワーク相互接続が作成され、リンクされます。
- VPC-DIインスタンスはサービス用に設定されています。

IaaS 管理者は、サーバをセットアップしてインターコネクトし、ハイパーバイザの VM テンプレートまたはオーケストレーション ソフトウェアを使用して、一連の VM、DI ネットワーク、およびサービスレベル契約（SLA）の要件を満たすための冗長性設定を作成します。

VPC-DI インスタンスを展開するには、オペレータの展開要件に対応する詳細な構成計画が必要です。

## ブート シーケンス

StarOS は、事前にインストールされたディスクテンプレートを QCOW2 形式で使用して、各 VM にインストールされます。スロット番号は、ESC および OpenStack によって管理されます。スロット番号は、VM 設定の一部として割り当てられます。スロット番号は、インストール時に自動検出されます。ハイパーバイザからスロット番号を検出できるように、インストーライメージのインストールは完全に自動化されています。詳細については、ソフトウェアインストールおよびネットワーク展開 （22 ページ）を参照してください。

VM の vNIC の設定を制御する方法については、起動パラメータファイルの作成 （23 ページ）を参照してください 。

各 VM は再起動し、VPC-DI インスタンスへの参加を試みます。ブートローダーは、自動起動（スクリプト）、ネットワーク起動、または手動起動によってインスタンスを起動します。

仮想 BIOS が完了すると、VM はローカル vHDD から起動し、CFE（共通ファームウェア環境）を実行します。CFE は、インストール中に作成されたパラメータファイルがあるかどうか、vHDD を検索します。このファイルが検出されて正しく解析された場合、CFE は VM のタイプとスロット番号に応じて異なるパスを取得します。いずれの場合も、最初の vNIC がネットワークのインターフェイスになります。

### CF ブートシーケンス

CF は、起動シーケンス中に次の機能を実行します。

- 他の CF が動作しているかどうかを確認します（DI ネットワーク経由）。
- 他の CF が動作している場合は、その CF からの起動を試みます。
  - 他の CF からパラメータと起動イメージを取得しようとします。
  - 成功した場合は、起動イメージを転送して実行します。
- 他の CF が動作していないか、または起動に失敗した場合は、個別に起動します。

- 起動または設定の優先順位については、ローカル vHDD 上で boot.sys ファイルを検索して解析します。
- ユーザによって（管理ネットワークまたはローカル vHDD を介して）中断されない限り、boot.sys 内の命令により起動を実行します。

CF 上の CFE では、starfile（起動可能イメージ）を CF 管理 vNIC を介してピアから外部 HTTP サーバまたは TFTP サーバへ、あるいは vHDD 上のローカルファイルからダウンロードできます。これは、boot.sys と StarOS の **boot** CLI コマンドによって実行されます。

（注） HTTP と TFTP による起動は、VIRTIO と VMXNET3 インターフェイスタイプでのみサポートされています。

DI ネットワーク上のネットワークプロトコルによって、マスターの CF が決まります。次に、マスターシップが DI ネットワークを介して SF VM に伝達されます。

## SF ブートシーケンス

SF がその vHDD から起動します。次に、DI ネットワークを介してアクティブな CF に接続し、正しいソフトウェアバージョンを起動したかどうかを判断します。SF が正しいソフトウェアバージョンを起動しなかった場合は、CF から正しいバージョンを転送し、自身を再起動します。ソフトウェアバージョンが正常に起動すると、ブートシーケンスが完了します。

# 帯域幅の要件

オペレータ展開ごとに、VPC-DI インスタンスをホストする L2 スイッチ上での帯域幅要件のモデリングが必要です。

主要なベアラートラフィックの他に、DI ネットワークは VM 間でセッションシグナリングと内部制御データも渡します。

内部制御トラフィックは冗長動作中は多くなりますが、通常動作時は大幅に低下します。次の場合に制御トラフィックの使用率が高くなります。

- アクティブ SF VM からスタンバイ SF へのタスクの移行
- スタンバイ SF の起動または再起動
- SF の起動または再起動
- SF またはスタンバイ CF の起動または再起動
- 大量のシグナリングトラフィック（1 秒あたりのコールイベント（CEP）レートが高い）
- CLI またはバルク統計情報の使用率が非常に多い

CEPS レート、設定、および管理操作に応じて、ベアラースループットに関係なく、各 VM は DI ネットワークインターフェイスに負荷をかけます。この負荷はばらつきが大きい可能性が

ありますが、平均して VM あたり 1 Gbps 未満です（他の VM よりも使用率が高い VM もあります）。

# キャパシティ、CEPS、およびスループット

VPC-DI 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。

最初のソフトウェアバージョンは、最大 2 つの CF と 14 の SF をサポートしています。

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。

ソフトウェアベースの送信バッチ処理により、システムパフォーマンスが大幅に向上します。

# 診断およびモニタリング

VPC-DI は VM 内で実行されるため、ハードウェア診断またはモニタリングは提供されません。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。基盤となる VM の設定を決定するには、VM ハードウェアの検証 （102 ページ）を参照してください。

VPC-DI は、CLI の **show** コマンド、バルク統計情報、MIB トラップなどの既存のメカニズムを使用して、VM ごとの vCPU、vRAM、および vNIC の使用状況をモニタおよびエクスポートします。ただし、オペレータは、ハイパーバイザのホストごとの物理 CPU、RAM、および NIC の値のモニタリングがより有用であることを確認できる場合があります。

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 VM から取得できますが、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ているため、パススルー PF NIC には適用されません。

# Cisco Prime Analytics

モビリティ向けの Cisco Prime の分析スイートには、VPC-DI インスタンスの拡張性管理が備わっています。

モビリティ向けの Cisco Prime は、次をサポートしています。

- 無線アクセスネットワーク（RAN）バックホールとパケットコア全体の統合されたオペレータワークフロー
- 一元化されたネットワークの可視性と高度なトラブルシューティングおよび診断

- 事前に統合されたネットワーク管理ソフトウェアコンポーネントによる統合に要する時間とリソースの削減

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。

# StarOS VPC-DI ビルドのコンポーネント

次の StarOS ビルドファイル名タイプが VPC-DI と関連付けられています。

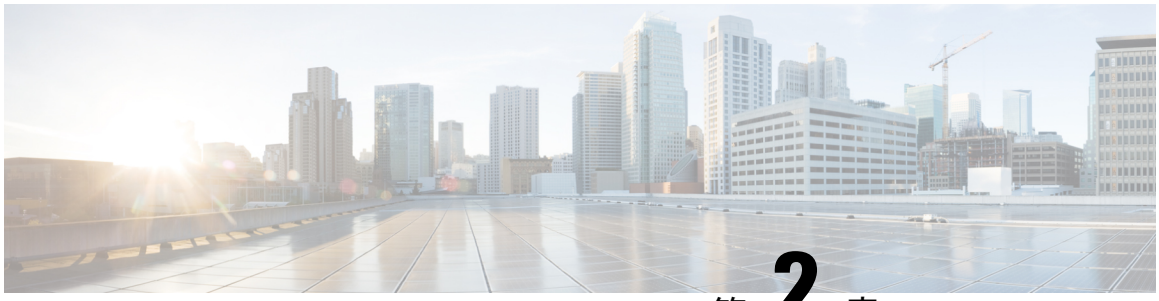
- **.qvpc-di-<version>.iso**：初期のインストールまたはスタートオーバー ISO ファイル。
- **.qvpc-di-<version>.bin**：更新、アップグレード、またはリカバリファイル（すでに実行しているシステム用）。
- **.qvpc-di-template-libvirt-kvm-<version>.tgz**：KVM libvirt テンプレートと ssi_install.sh。
- **.qvpc-di.qcow2.tgz**：KVM QCOW2 ディスクテンプレート。
- **.qvpc-di-template-vmware.tgz**：VMware ファイル。
- **.qvpc-di-template-vmware-<version>.ova**：VMware OVA テンプレート。

# 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-DI のコンポーネントが正しくインストールされ、市販（COTS）サーバ上の仮想マシン（VM）上で実行されていることを前提としています。詳細については、プロビジョニング （19 ページ）を参照してください。

また、DI ネットワークも、DI ネットワーク （6 ページ） と 帯域幅の要件 （20 ページ） で指定した要件を満たすため、データセンター内でプロビジョニングされている必要があります。

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-DI 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

# VPC-DI 設置上の注意事項

このガイドでは、VPC-DIのコンポーネントが正しくインストールされ、市販（COTS）サーバ上の仮想マシン（VM）上で実行されていることを前提としています。この章では、インストールプロセスに役立ついくつかのインストールの注意事項について説明します。



## 起動パラメータファイルの作成

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。通常、パラメータは、StarOS を正常にロードし、仮想スロット番号、VM のタイプ、NIC の割り当て、ネットワークボンディングの設定などの項目を指定するために必要です。

デフォルトでは、VPC-DI 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。特定の順序に従って手動でvNICsを設定するには、起動パラメータファイルを作成する必要があります。また、VNFMインターフェイスを有効にする場合も、起動パラメータファイルを作成する必要があります。

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 ドライブの最初のパーティション（たとえば、 */boot1/param.cfg*）にあります。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競合がある場合、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を作成しない場合は、デフォルトファイルが使用されます。起動パラメータファイルを作成する場合は、起動パラメータの設定 （29 ページ） で説明されているすべてのパラメータを定義する必要があります。

# 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 ロール

ネットワーク インターフェイスは、VM が CF または SF に使用されているかどうかに応じて、特定の役割を果たします。

すべてのシステム VM には、DI 内部ネットワークへのネットワーク インターフェイス接続が備わっています。このネットワークは、VPC-DI インスタンス内のすべての VM 同士をリンクします。このネットワークは VPC-DI インスタンスに非公開とする必要があります。また、このネットワークはシステムソフトウェアによって設定されます。

仮想ネットワーク機能（VNF）マネージャ（VNFM）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すべての 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、VNFM 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。

CF では、1 つの追加インターフェイスが管理ネットワーク インターフェイスに接続します。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。

SF では、追加の 0 ～ 12 個のネットワーク インターフェイスが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。

**表 *2:* ネットワーク インターフェイス ロール**

| インターフェイス ロール | 説明 |
|---|---|
| DI_INTERFACE | VM のすべてのタイプに必要な DI 内部ネットワークへのインターフェイス |

| インターフェイス ロール | 説明 |
|---|---|
| MGMT_INTERFACE | CF VM 上の管理ポートへのインターフェイス |
| SERVICE#_INTERFACE | SF VM のサービスポート番号 #（# は 1 ～ 12） |
| VNFM_INTERFACE | 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（VM のすべてのタイプで有効） |

（注） VIRTIO のインターフェイスは DI_INTERFACE ロールと SERVICE#_INTERFACE ロールに使用できますが、推奨されません。

# ネットワーク インターフェイス ID

デフォルトでは、VPC-DI VM によって検出された最初の NIC には、DI 内部ネットワークロールが割り当てられます。追加ポートは、SF の CF またはサービスポートの管理インターフェイスとして機能します。デフォルトでは、インターフェイスは VNFM インターフェイスとして使用されません。

VPC-DI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-DI 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、PCI ID、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 *3 :* 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 メモリバルーニング | 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 *4*: 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<br>0x8086 / 0x10c6<br>0x8086 / 0x10c7<br>0x8086 / 0x10c8<br>0x8086 / 0x150b<br>0x8086 / 0x10dd<br>0x8086 / 0x10ec<br>0x8086 / 0x10f1<br>0x8086 / 0x10e1<br>0x8086 / 0x10db<br>0x8086 / 0x1508<br>0x8086 / 0x10f7<br>0x8086 / 0x10fc<br>0x8086 / 0x1517<br>0x8086 / 0x10fb<br>0x8086 / 0x1507<br>0x8086 / 0x1514<br>0x8086 / 0x10f9<br>0x8086 / 0x152a<br>0x8086 / 0x1529<br>0x8086 / 0x151c<br>0x8086 / 0x10f8<br>0x8086 / 0x1528<br>0x8086 / 0x154d<br>0x8086 / 0x154f<br>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<br>0x8086 / 0x1515 | ixgbevf |
| Cisco UCS NIC | 0x1137 / 0x0043<br>0x1137 / 0x0044<br>0x1137 / 0x0071 | enic |
| Mellanox ConnectX-5<br>（注） Mellanox は、ユーザプレーンでのみサポートされています。 | 0x15b3 / 0x1017<br>0x15b3 / 0x1018 | mlx5_core |
| Intel XL 710 ファミリ NIC（PF） | 0x8086 / 0x1572（40 ギガ）<br>0x8086 / 0x1574（40 ギガ）<br>0x8086 / 0x1580（40 ギガ）<br>0x8086 / 0x1581（40 ギガ）<br>0x8086 / 0x1583（40 ギガ）<br>0x8086 / 0x1584（40 ギガ）<br>0x8086 / 0x1585（40 ギガ）<br>0x8086 / 0x158a（25 ギガ）<br>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-iov-config-guide-gbe-linux-brief.pdf

# 起動パラメータの設定

起動パラメータファイルを作成しない場合は、デフォルトファイルが使用されます。起動パラメータファイルを作成する場合は、このタスクで説明されているすべてのパラメータを定義する必要があります。

始める前に

VM インターフェイスにおけるインターフェイス識別子の決定の詳細については、ネットワーク インターフェイス ロール （24 ページ）および ネットワーク インターフェイス ID （25 ページ） を参照してください。

**ステップ 1** **CARDSLOT**=*slot-number*

*slot_number* は、スロット番号または VM を示す 1 ～ 32 の整数です。CF スロットには 1 または 2 を指定できます。SF スロットの範囲は 3 ～ 48 です。

**ステップ 2** **CARDTYPE**=*card-type*

*card-type* は、VM が CF か SF かを識別します。

- コントロール機能には、0x40010100 を使用します。
- サービス機能には、0x42020100 を使用します。

**ステップ 3** *interface-role*_**INTERFACE**=*interface-id*

*interface-role* の有効な値は次のとおりです。

- DI
- MGMT
- SERVICE#、# の範囲は 1 ～ 12 です。
- VNFM

たとえば、DI_INTERFACE=interface-id です。

インターフェイスロールの詳細については、ネットワーク インターフェイス ロール （24 ページ）を参照してください。

*interface-id* の有効な値は次のとおりです。

- MAC: xx:xx:xx:xx:xx:xx
- PCI_ID:xxxx:xx:xx.x (Domain:Bus:Device.Function)
- TYPE:*drive-name–instance-number*
- BOND: *slave-interface-A,slave-interface-B*

インターフェイス識別子の決定については、ネットワーク インターフェイス ID （25 ページ）を参照してください。

**例**：

この例では、MAC アドレスでインターフェイスを識別します。

```
DI_INTERFACE=MAC:00:01:02:03:04:05
```

この例では、ゲスト PCI アドレスでインターフェイスを識別します。

```
DI_INTERFACE=PCI_ID:0000:01:02.0
```

この例では、インターフェイスのタイプ（1 番目の virtio インターフェイス）でインターフェイスを識別します。

```
DI_INTERFACE=TYPE:enic-1
```

**例：**

この例では、インターフェイスをネットワーク ボンド インターフェイスとして識別します。次に、MAC アドレス、PCI 識別子、およびインターフェイスタイプを使用してインターフェイスを識別する例を示します。

```
DI_INTERFACE=BOND:MAC:00:01:02:03:04:05,MAC:00:01:02:03:04:06
# or
DI_INTERFACE=BOND:PCI_ID:0000:01:01.0,PCI_ID:0000:01:02.0
# or
DI_INTERFACE=BOND:TYPE:enic-1,TYPE:enic-2
```

# ネットワーク インターフェイス ボンディングの設定

システムは、ネットワークインターフェイスのペアをアクティブとスタンバイの結合インターフェイスに設定することをサポートしています。一度に 1 つのインターフェイスだけがアクティブになり、障害の検出は物理リンクの損失に限定されます。このタスクを使用して、結合インターフェイスを設定します。

すべてのボンディングの変数名は、*interface-role*_BOND 形式を使用します。インターフェイスロールについては、ネットワーク インターフェイス ロール （24 ページ） を参照してください。

### 始める前に

このタスクで説明されているすべての起動パラメータはオプションです。これらのパラメータが必要な場合は、起動パラメータの設定 （29 ページ） で説明されている必須のパラメータとともに、起動パラメータファイルに追加してください。

**ステップ 1** *interface-role*_BOND_PRIMARY=*interface-id*

特定のインターフェイスが時間の大半をアクティブにするように設定している場合は、プライマリ スレーブ インターフェイスを設定します。デフォルトの結合設定では、プライマリスレーブは選択されません。

インターフェイスロールについては、ネットワーク インターフェイス ロール （24 ページ） を参照してください。インターフェイス識別子については、ネットワーク インターフェイス ID （25 ページ） を参照してください。

（注） デフォルトでは、復帰ポリシーは、新しいアクティブリンクの後続の障害時に、ボンドがプライマリインターフェイスに戻るだけというものです。

デフォルトでは、障害検出方法は、ボンドがドライバ状態を使用して、基本となるインターフェイスのリンク状態をポーリングすることです。

例：

次の例では、MAC アドレスを使用してプライマリインターフェイスを指定します。

```
DI_INTERFACE_BOND_PRIMARY=MAC:00:01:02:03:04:05
```

次に、PCI 識別子を使用してプライマリインターフェイスを指定する例を示します。

```
DI_INTERFACE_BOND_PRIMARY=BOND:PCI_ID:0000:01:01.0
```

次の例では、インターフェイスタイプの識別子を使用してプライマリインターフェイスを指定します。

例：

```
DI_INTERFACE_BOND_PRIMARY=BOND:TYPE:enic-1
```

**ステップ 2** *interface-role*_BOND_MII_POLL = *poll-interval*

MII がリンク検出に使用される場合に使用するポーリング間隔（ミリ秒単位）を指定します。ポーリング間隔は 0 ～ 1000 の範囲で指定できます。デフォルトは 100 です。

**ステップ 3** *interface-role*_BOND_MII_UPDELAY=*slave-enable-delay*

リンクの検出に MII が使用されている場合は、リンクの障害後にスレーブインターフェイスを有効にする前にリンクが安定するまでの待機時間を指定します。リンク状態は、最初に検出されたときにバウンスできます。この遅延により、インターフェイスを使用する前にリンクを安定させることができます。これにより、結合インターフェイスに対してアクティブなスレーブの過剰なフリップが回避されます。

スレーブの有効化遅延は、MII ポーリング間隔の倍数である必要があります。値はミリ秒単位で、デフォルトは 0 です。

**ステップ 4** *interface-role*_BOND_MII_DOWNDELAY=*slave-disable-delay*

オプションです。この機能を使用すると、リンク検出に MII が使用されている場合に、スレーブインターフェイスがダウンしていることを宣言する前に、ボンドが待機するようになります。スレーブの無効化遅延は、MII ポーリング間隔の倍数である必要があります。値はミリ秒単位で、デフォルトは 0 です。

# VNFM インターフェイスの設定

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。このインターフェイスはメインのアプリケーションの前に起動され、起動パラメータのみを使用して設定できます。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 インターフェイスを設定するには、次のタスクを実行します。

### 始める前に

このタスクで説明されているすべての起動パラメータはオプションです。これらのパラメータが必要な場合は、起動パラメータの設定 （29 ページ） で説明されている必須のパラメータとともに、起動パラメータファイルに追加してください。

**ステップ 1** VNFM_IPV4_ENABLE={true | false}

VNFM インターフェイスを有効にします。

**ステップ 2** VNFM_CARTRIDGE_AGENT={true | false}

カートリッジエージェントを有効にします。VNFM がカートリッジエージェントを使用している場合は、これを有効にする必要があります。

**ステップ 3** VNFM_IPV4_DHCP_ENABLE={true | false}

VNFM 上で DHCP を有効にします。

**ステップ 4** VNFM_IPV4_ADDRESS=*x.x.x.x*

DHCP が使用されていない VNFM の IP アドレスを指定します。

**ステップ 5** VNFM_IPV4_NETMASK=*x.x.x.x*

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。

**ステップ 6** VNFM_IPV4_GATEWAY=*x.x.x.x*

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。

## VNFM インターフェイスオプション

（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す。

| 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す。

### 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:x:x |
| VNFM_IPV6_GATEWAY | x:x:x:x:x:x:x:x |

# DI ネットワーク VLAN の設定

DI ネットワークには、使用可能な一意の分離ネットワークが必要です。パススルーインターフェイスを使用する場合、カスタマーネットワークでの VPC-DI インスタンスの分離を容易にするために、VLAN ID を設定できます。オプションとして、ホストに専用ポートがある場合は、DI ネットワーク VLAN をホストまたは L2 スイッチでタグ付けすることもできます。

VLAN を設定するには、このタスクを実行します。

### 始める前に

このタスクで説明されているすべての起動パラメータはオプションです。これらのパラメータが必要な場合は、起動パラメータの設定 （29 ページ） で説明されている必須のパラメータとともに、起動パラメータファイルに追加してください。

DI_Internal_VLANID=*vlan-id*

内部 DI ネットワークの VLAN ID を指定します。値の範囲は 1 ～ 4094 です。

**例**：

```
DI_INTERNAL_VLANID=10
```

# IFTASK の調整可能なパラメータの設定

デフォルトでは、DPDK は CPU コアの 30% を内部フォワーダタスク （IFtask）のプロセスに割り当てます。これらの起動パラメータを使用して、IFTASK に割り当てられたリソースを設定できます。IFTASK の CPU コア割り当てに関する情報を表示するには、**show cpu info** と **show cpu verbose** コマンドを使用します。

（注） これらは、細心の注意を払って設定する必要があるオプションのパラメータです。

**ステップ 1** （オプション）**IFTASK_CORES=***percentage-of-cores*

IFTASK に割り当てる CPU コアの割合を指定します。値の範囲は 0 ～ 100パーセントです。デフォルトは 30 です。

**ステップ 2** （オプション）**MCDMA_THREAD_DISABLE=***percentage-of-iftask-cores*

MCDMA と VNPU の分割を使用するのではなく、すべてのコアで PMD を実行するには、MCDMA_THREAD_DISABLE パラメータを 1 に設定します。

**ステップ 3** （オプション）**IFTASK_SERVICE_TYPE=***value*

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。次のサービスタイプを指定できます。

- 0 = VPC サービスタイプ
- 1 = GiLAN サービスタイプ
- 2 = ePDG サービスタイプ
- 3 = CUPS コントローラサービスタイプ
- 4 = CUPS フォワーダサービスタイプ

デフォルトは 0 です。

**ステップ 4** （オプション）**IFTASK_CRYPTO_CORES=***value*

**IFTASK_SERVICE_TYPE** が「2」（EPDG）に設定されている場合、このパラメータは、暗号処理に割り当てる iftask コアの割合を指定します。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。

（注） 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してください。他のサービスタイプに設定されている場合は、このパラメータを「0」に設定する必要があります。

**ステップ 5** （オプション）**IFTASK_DISABLE_NUMA_OPT=***value*

複数の NUMA ノードがホストによって VM に提供されている場合でも、この設定を使用して NUMA 最適化を無効にします。このオプションは、NUMAの最適化が何らかの理由で望ましくない場合に設定できます。

- NO = 有効（デフォルト）
- YES = 無効

NUMA の最適化は、次の場合を除き、デフォルトで有効になっています。

- NUMA ノードとセルの数が 2 に等しくない。
- カードタイプが制御機能（CF）、アプリケーション機能（AF）、またはネットワーク機能（NF）になっている。サービス機能（SF）VM のみが NUMA をサポートしている。
- VM のサービスタイプが VPC ではない。NUMAが VPC サービスタイプでのみサポートされている。
- この設定が、明示的に [YES]（**IFTASK_DISABLE_NUMA_OPT=YES**）に設定されている。

**ステップ 6** （オプション）**IFTASK_VNPU_TX_MODE=***value*

Ultra M 展開のコンピューティングノードには、28 個のコアがあります。これらのコアのうち 2 つは、ホストで使用するために予約されています。26 個のコアを使用すると、MCDMA 機能を実行するために使用されるコア間で MCDMA チャネルが均等に分散されます。

この設定を有効にすると、iftask 内の MCDMA 機能のコアは MCDMA コアおよび VNPU TX ルックアップコアとして均等に分割されます。

- 0 = 無効（デフォルト）
- 1 = 有効

**ステップ 7** （オプション）**MULTI_SEG_MBUF_ENABLE=***value*

リリース 21.6 以降のデフォルトでは、システムは Ixgbe pf/vf ドライバのすべてのメモリプールで、サイズの小さいバッファを使用したマルチセグメントの送受信の使用を有効にします。この機能により、IFTASK の全体的なメモリサイズが削減され、小規模な展開に適したものになります。

- 1 = true（xgbe NIC の場合はデフォルト）。
- 0 = false（他のすべての NIC の場合はデフォルト）。

**重要** デフォルトでは、この機能が有効になっているため、Ixgbe NIC を使用するシステムで 21.6 にアップグレードする場合は注意が必要です。

この機能は、Ixgbe vf/pf NICs NICs を使用していないシステムでは自動的に無効になります。

---

**例**

StarOS の **show cloud hardware iftask** *card_number* コマンドを使用して、起動パラメータが有効になったことを確認します。

```
[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
  Number of cores for Crypto        0
Hugepage size:               2048 kB
Total hugepages:          3670016 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
```

# 最大 IFtask スレッドサポートの改善

## 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 | 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 | *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 |
|---|---|
| このリリース以降、サポートされる IFtask スレッド設定の最大数は 22 コアに増えました。 | 21.8 |
| 最初の導入。 | 21.2 よりも前 |

## 変更された機能

設定されている（/tmp/iftask.cfg 内）DPDK 内部フォワーダ（IFTask）スレッドの数が 14 コアを超えている場合、IFTask はパケットをドロップするか、またはエラーを表示します。

**以前の動作**：現在、IFtask スレッド設定の最大数は 14 コアのみに限定されています。

**新しい動作**：リリース 21.8 以降、サポートされる IFtask スレッドの最大数が 22 コアに増加しました。

# MTU サイズの設定

デフォルトでは、IFTASK プロセスは最大インターフェイス MTU を次のように設定します。

- サービスインターフェイス：2,100 バイト

• DI ネットワーク インターフェイス：7,100 バイト

これらのデフォルトを変更するには、param.cfg ファイルで次のパラメータを設定します。

**表 *5 : MTU* サイズパラメータ**

| パラメータ名 | 範囲 | デフォルト値 |
|---|---|---|
| **DI_INTERFACE_MTU=** | 576-9100 | 7100 |
| **SERVICE_INTERFACE_MTU=** | 576-9100 | 2100 |

ジャンボフレームをサポートしていないシステムの MTU サイズの設定については、サポート対象 MTU を超えるトラフィックのサポートの設定 （39 ページ）を参照してください。

# サポート対象 MTU を超えるトラフィックのサポートの設定

デフォルトでは、システムが動作するにはジャンボフレームのサポートが必要です。インフラストラクチャがジャンボフレームをサポートしていない場合でも、システムを実行できます。ただし、起動パラメータファイルで DI 内部ネットワークの MTU を 1500 に指定する必要があります。これにより、IFTASK が、サポートされている MTU を超える DI ネットワークトラフィックを処理できるようになります。

### 始める前に

このタスクで説明されているすべての起動パラメータはオプションです。これらのパラメータが必要な場合は、起動パラメータの設定 （29 ページ） で説明されている必須のパラメータとともに、起動パラメータファイルに追加してください。

DI_INTERFACE_MTU=*1500*

ソフトウェアがジャンボフレームを適切に処理できるように、DI内部ネットワークがジャンボフレームをサポートしないことを指定します。

# 起動パラメータファイルの例

この例では、2 つの VIRTIO インターフェイスを備えたスロット 1 の CF の起動パラメータファイルを示しています。

```
CARDSLOT=1
CARDTYPE=0x40010100
DI_INTERFACE=TYPE:enic-1
MGMT_INTERFACE=TYPE:virtio_net-2
```

この例では、3 つの VIRTIO インターフェイスを備えたスロット 3 の SF の起動パラメータファイルを示しています。

```
CARDSLOT=3
CARDTYPE=0x42020100
DI_INTERFACE=TYPE:enic-1
SERVICE1_INTERFACE=TYPE:enic-3
SERVICE2_INTERFACE=TYPE:enic-4
```

この例では、パススルー NIC、ボンディングが設定された CF、および VLAN 上に DI 内部ネットワークがある CF の起動パラメータファイルを示しています。

```
CARDSLOT=1
CARDTYPE=0x40010100
DI_INTERFACE=BOND:TYPE:enic-1,TYPE:enic-2
MGMT_INTERFACE=BOND:TYPE:ixgbe-3,TYPE:ixgbe-4
DI_INTERNAL_VLANID=10
```

# VPC-DI ESC を使用したオンボーディング

ESC を使用して VPC-DI のインスタンスを開始できます。

## OpenStack での ESC を使用した VPC-DI のオンボーディング

この手順では、OpenStack 環境で ESC のインスタンスに VPC-DI をオンボードする方法について説明します。

### 始める前に

この手順では、ネットワークアクセスを使用してリリース Juno 以降を実行している動作中の OpenStack 環境内に ESC が作成されていることを前提としています。ESC の詳細なインストール手順については、『*Cisco Elastic Services Controller 2.3 Install and Upgrade Guide*』（http://www.cisco.com/c/en/us/td/docs/net_mgmt/elastic_services_controller/2-3/install/guide/Cisco-Elastic-Services-Controller-Install-Upgrade-Guide-2-3.html）を参照してください。ESC 設定の説明については、『*Cisco Elastic Services Controller 2.3 User Guide*』（http://www.cisco.com/c/en/us/td/docs/net_mgmt/elastic_services_controller/2-3/user/guide/Cisco-Elastic-Services-Controller-User-Guide-2-3.html）を参照してください。

このリリースでサポートされている Elastic Services Controller のバージョンを確認するには、『リリースノート』を参照してください。

**ステップ 1** CF および SF の VPC-DI インスタンスの qcow イメージを取得します。

イメージを含む tarball ファイルには、リリース番号に応じて `production.xxxxx.qvpc-di.qcow2.tgz` というような名前が付けられています。このアーカイブには、CF と SF それぞれの 2 つのイメージ（qvpc-di-cf.qcow2 と qvpc-di-xf.qcow2）が含まれています。

**ステップ 2** VPC-DI コマンドを使用すると **glance image-create** イメージが瞬時に作成されます。

**例**：

```
$ glance image-create --file qvpc-di-cf.qcow2 --container-format bare --disk-format
    qcow2 --is-public true --name cisco-qvpc-cf

$ glance image-create --file qvpc-di-xf.qcow2 --container-format bare --disk-format
    qcow2 --is-public true --name cisco-qvpc-xf
```

**ステップ 3** サンプルの VPC-DI の初期化 tarball（vpc_esc_sample.tgz）を取得します。

**ステップ 4** サンプルの VPC-DI の初期化 tarball を管理者ホーム（/home/admin/）の ESC VM にコピーします。

**ステップ 5** サンプルの VPC-DI 初期化 tarball を vnf ディレクトリ（opt/cisco/vnfs/cisco-qvpc/）に解凍します。

**ステップ 6** コマンドライン API コマンド **esc_nc_cli edit-config** を使用して、アーティファクトを作成し ます。

**例**：

```
esc_nc_cli edit-config  /opt/cisco/vnfs/cisco-qvpc/dep/artifacts.xml
```

**ステップ 7** **esc_nc_cli edit-config** コマンドを使用して VPC-DI を展開します。

（注） ESC を再展開する前に、既存の展開を必ず削除してください。手順 10 を参照してください。

VPC-DI は、`vnfs/cisco-qvpc/dep/`にある dep.xml ファイルを使用して展開されます。一般に、デフォルトの dep.xml ファイルを使用できます。展開をカスタマイズする必要がある場合は、このファイルに必要な変更を加えます。たとえば、VPC-DI のシャーシ ID の作成に使用するシャーシキーを編集するには、dep.xml ファイルの該当するセクションを編集します。

```
<property>
  <name>CHASSIS_KEY</name>
  <value>164c03a0-eebb-44a8-87fa-20c791c0aa6d</value>
</property>
```

dep.xml ファイルの詳細については、次を参照してください。 http://www.cisco.com/c/en/us/td/docs/net_mgmt/elastic_services_controller/2-2/deployment/Cisco-Elastic-Services-Controller-2-2-Deployment-Attributes.pdf

**例**：

```
esc_nc_cli edit-config /opt/cisco/vnfs/cisco-qvpc/dep/dep.xml
```

このコマンドは、テナントコアがまだ存在していない場合に新しいテナントコアを作成できます。使用されている dep*.xml によっては、展開で「クォータ超過」エラーが発生する可能性があります。すべての SF と CF が起動しない場合は、OpenStack でデフォルトのテナントクォータとテナントコアのクォータを確認します。これを実行するコマンドは `$ nova quota-defaults; nova quota-show --tenant Core` です。

**ステップ 8** ログファイルの `var/log/esc/yangesc.log`で展開ステータスを確認します。

**ステップ 9** VPC-DI が統合されるまで待ちます。

**ステップ 10** VPC-DI の展開を削除するには、ESC CLI コマンドの **esc_nc_cli delete-dep** *Tenant deployment-name* を使用します。

**例**：

```
$ ./esc_nc_cli delete-dep Core cisco-qvpc
```

**ステップ 11** （オプション）カスタムモニタリングを使用するには、VPC-DI を展開する（手順 7）前に次の手順を実行します。

a) mib ファイルをコピーします。

**例：**

```
sudo cp /opt/cisco/vnfs/cisco-qvpc/config/starent.my /usr/share/snmp/mibs/
```

b) VPC-DI のダイナミック マッピング メトリックを既存のマッピングに追加します。ファイル /opt/cisco/vnfs/cisco-qvpc/config/dynamic_mappings_snippet.xml の内容を /opt/cisco/esc/esc-dynamic-mapping/dynamic_mappings.xml にマージします。この手順を実行するには、細心の注意が必要です。SF ごとに、opt/cisco/esc/esc-dynamic-mapping/dynamic_mappings.xml に 1 つのダイナミックマッピングが必要です。

# ESC を使用した VPC-DI のオンボーディングのカスタマイズ

ESC を使用して VPC-DI をオンボードすると、VPC-DI ファイルは /opt/cisco/vnfs/cisco-qvpc/ ディレクトリに配置されます。VPC-DI システムのインストールにはさまざまな変更を加えることができます。次のファイルは必要に応じて変更できます。

### VNF 展開ファイル

展開ファイルは、ディスクの /opt/cisco/vnfs/cisco-qvpc/dep/ にコピーされます。

必要に応じて、追加のファイルを作成できます。

### 起動パラメータファイル

ESC を使用した VPC-DI のオンボードの場合は、ディスク上に配置されたさまざまな CF VM と SF VM 用にいくつかの異なる起動ファイルが用意されています。これらのファイル /opt/cisco/vnfs/cisco-qvpc/config/ にコピーされます。

- param.cfg
- param_sf.cfg

これらは、必要に応じてカスタマイズできます。起動パラメータのファイル形式の詳細については、起動パラメータファイルの作成 （23 ページ）を参照してください。

次に、CF の起動パラメータファイルの例を示します。

```
CARDSLOT=$SLOT_CARD_NUMBER
CPUID=0
CARDTYPE=$CARD_TYPE_NUM

DI_INTERFACE=TYPE:enic-1
VNFM_INTERFACE=TYPE:virtio_net-2
MGMT_INTERFACE=TYPE:virtio_net-3

VNFM_IPV4_ENABLE=true
VNFM_IPV4_DHCP_ENABLE=true
```

次に、SF の起動パラメータファイルの例を示します。

```
CARDSLOT=$SLOT_CARD_NUMBER
CPUID=0
CARDTYPE=$CARD_TYPE_NUM

DI_INTERFACE=TYPE:enic-1
VNFM_INTERFACE=TYPE:enic-2
SERVICE1_INTERFACE=TYPE:enic-3

VNFM_IPV4_ENABLE=true
VNFM_IPV4_DHCP_ENABLE=true
```

### 設定ファイル

/cisco/images/system.cfg にある設定ファイルはカスタマイズできます。次に、標準設定ファイルの例を示します。

```
config
system hostname $VPC_HOSTNAME
clock timezone $TIMEZONE
context local
administrator admin password $ADMIN_PASS ftp

interface LOCAL1
ip address $CF_VIP_ADDR $CF_VIP_NETMASK
ip route 0.0.0.0 0.0.0.0 $NICID_1_GATEWAY LOCAL1
ip domain-lookup
ip domain-name $CF_DOMAIN_NAME
ip name-servers $CF_NAME_SERVER
ssh generate key
server sshd
subsystem sftp
port ethernet 1/1
bind interface LOCAL1 local
no shutdown
snmp community $SNMP_COMMUNITY read-only
end
```

詳細については、コンフィギュレーション ファイルの概要 （72 ページ）を参照してください。

# OpenStack のパフォーマンスの最適化

Cisco ESC では、Non-Uniform Memory Access（NUMA）ノード設定などの OpenStack Kilo リリースを使用して、大きなページのサポートやゲスト vCPU のピン接続など、数多くのハイパーバイザ最適化を実行できます。

- vCPU ピン接続：ゲスト vCPU をホストの一連の物理 CPU に正確に固定する機能です。これにより、vCPU が物理リソースを使用できるようになるまで待機することを未然に防ぎます。
- 大規模ページ：仮想リソースに大きなメモリブロックを割り当てます。

- PCI ベースの NUMA のスケジューリング：OpenStack 内のインスタンスに PCI デバイスを割り当てる機能です。

（注） これらの OpenStack パフォーマンスの最適化は OpenStack Kilo バージョンのみを使用してサポートされます。

OpenStack Kilo はさまざまなメカニズムを使用してさまざまなハードウェア アクセラレーション機能を提供します。このようなメカニズムの 1 つには、フレーバオブジェクトのキー値属性の設定が含まれます。特定のフレーバを参照する VM インスタンス化要求は、対応するハードウェア アクセラレーション機能を効率的に要求します。必要な OpenStack 設定が OpenStack の制御ノードとコンピューティングノードに配置されている場合は、OpenStack コンピューティングサービス（「nova」）が適切なコンピューティングノードを選択し、対応するリソースを割り当てます。

VPC-DI でハードウェア アクセラレーションを使用するには、Cisco ESC に新しいフレーバを作成し、NETCONF インターフェイスか REST インターフェイスを使用して必要なメタデータ属性を追加する必要があります。フレーバの詳細については、『*Cisco Elastic Services Controller User Guide*』（http://www.cisco.com/c/en/us/support/cloud-systems-management/elastic-services-controller-2-1/model.html）の「*Managing Flavors*」を参照してください。また、多数の OpenStack 設定を変更する必要がある場合もあります。ハードウェア アクセラレーションを実装する手順については、次のタスクで説明します。

（注） メタデータの追加は、既存のものではなく、新しいフレーバでのみサポートされています。メタデータ属性を既存のフレーバに追加する必要がある場合は、OpenStack と直接連携する必要があります。

## CPU ピニングの設定

**ステップ 1** 各 OpenStack 制御ノードで、スケジューラを設定します。

a) NUMAToplogy フィルタと AggregateInstanceExtraSpec フィルタを含めるようにスケジューラフィルタを設定します。

**例**：

```
$ sudo vim /etc/nova/nova.conf
...
scheduler_default_filters=RetryFilter,AvailabilityZoneFilter,RamFilter,ComputeFilter,
ComputeCapabilitiesFilter,ImagePropertiesFilter,CoreFilter,NUMATopologyFilter,
AggregateInstanceExtraSpecsFilter
```

b) nova スケジューラサービスを再起動します。

**例**：

```
$ sudo systemctl restart openstack-nova-scheduler.service
```

**ステップ 2** 関連する各 OpenStack のコンピューティングノードで、どのハイパーバイザプロセスをゲストに使用し、どのハイパーバイザプロセスをゲストに使用しないようにするかを設定します。

a) ハイパーバイザプロセスがゲスト用に予約されているコアで実行されていないことを確認します。

たとえば、ゲストにコア 2、3、6、および 7 を予約するには、grub ブートローダーを更新して再起動します。

**例**：

```
$ sudo grubby --update-kernel=ALL --args="isolcpus=2,3,6,7"
$ sudo grub2-install /dev/sda
$ sudo reboot
```

b) カーネルのコマンドラインに変更が反映されていることを確認します。

**例**：

```
$ cat /proc/cmdline
...  isolcpus=2,3,6,7 ...
```

c) ゲスト仮想マシンのインスタンスを設定して、特定のコアでの実行のみが許可され、ハイパーバイザプロセス用の RAM を予約するようにします。たとえば、コア 2、3、6、および 7 を使用し、ハイパーバイザプロセスに 512 MB を予約するには、次の手順を実行します。

**例**：

```
$ sudo vim /etc/nova/nova.conf
...
vcpu_pin_set=2,3,6,7
...
reserved_host_memory_mb=512
...
```

d) nova コンピューティングサービスを再起動します。

**例**：

```
$ sudo systemctl restart openstack-nova-compute.service
```

**ステップ 3** グローバルパラメータを設定します。

a) ピン接続要求を受信したホストに対して `performance-pinned` ホスト集約を作成し、その識別のために任意の属性 `pinned=true` を追加します。

**例**：

```
$ nova aggregate-create performance-pinned
$ nova aggregate-set-metadata performance-pinned pinned=true
```

b) 他のすべてのホストに対して通常の集約を作成し、同じ任意の属性を追加しますが、それを識別するには、`pinned=false` を設定します。

**例**：

```
$ nova aggregate-create normal
$ nova aggregate-set-metadata normal pinned=false
```

c) 以前に有効になっていたコンピューティングノードを、performance-pinned ホスト集約に追加し、他のすべてのコンピューティングノードを通常のホスト集約に追加します。

**例**：

```
$ nova aggregate-add-host normal             compute100.cloud.com
$ nova aggregate-add-host normal             compute101.cloud.com
$ nova aggregate-add-host normal             compute102.cloud.com
$ nova aggregate-add-host performance-pinned compute103.cloud.com
$ nova aggregate-add-host performance-pinned compute104.cloud.com
$ nova aggregate-add-host performance-pinned compute105.cloud.com
```

**ステップ 4** Cisco ESC ノースバウンド API を使用してフレーバー属性を設定します。

a) フレーバー属性を設定します。

- hw:cpu_policy=dedicated
- aggregate_instance_extra_specs:pinned=true

このフレーバーを使用して作成されたすべてのインスタンスは、集約メタデータ内のピン接続された pinned=true を使用して、ホスト集約のホストに送信されます。

**例**：

```
 version='1.0' encoding='ASCII'?>
<esc_datamodel xmlns="http://www.cisco.com/esc/esc">
  <flavors>
    <flavor>
      <name>testfl6</name>
      <vcpus>1</vcpus>
      <memory_mb>2048</memory_mb>
      <root_disk_mb>10240</root_disk_mb>
      <ephemeral_disk_mb>0</ephemeral_disk_mb>
      <swap_disk_mb>0</swap_disk_mb>
      <properties>
        <property>
          <name>hw:cpu_policy</name><value>dedicated</value>
          <name>aggregate_instance_extra_specs:pinned</name><value>true</value>
        </property>
      </properties>
    </flavor>
  </flavors>
</esc_datamodel>
```

b) 追加の仕様が通常の集約のコンピューティングホストと一致するように、他のすべてのフレーバーを更新します。

**例**：

```
$ nova flavor-key <flavor_Id> set aggregate_instance_extra_specs:pinned=false
```

c) 確認するには、変更されたフレーバーを使用して VM インスタンスを起動し、VM インスタンスが開始されたコンピューティングノードを検索します。

**例**：

```
$ nova boot --image <test-image> --flavor <modified-flavor> test-instance
$ nova show test-instance | egrep
'OS-EXT-SRV-ATTR:hypervisor_hostname|OS-EXT-SRV-ATTR:instance_name'
| OS-EXT-SRV-ATTR:hypervisor_hostname  | compute3.cloud.com
| OS-EXT-SRV-ATTR:instance_name        | instance-00000cee
```

d) 返されたコンピューティングノードにログインし、**virsh** ツールを使用して、返されたインスタンスの XML を抽出します。

**例**：

```
$ ssh compute3.cloud.com
...
$ sudo virsh dumpxml instance-00000cee
...
  <vcpu placement='static' cpuset='2-3,6-7'>1</vcpu>
```

## 大規模なページの設定

OpenStack 設定で大規模なページを設定するには、この手順を実行します。

**ステップ 1** ページ数を設定するには、/etc/sysctrl.conf を編集します。

**例**：

```
vm.nr_hugepages = 32768
```

**ステップ 2** ページサイズを設定するには、/proc/meminfo を編集します。

**例**：

```
Hugepagesize: 2048 kB
```

**ステップ 3** Cisco ESC ノースバウンド API を使用して、hw: mem_page_size = 2048 のフレーバを作成します。

**例**：

```
<?xml version='1.0' encoding='ASCII'?>
<esc_datamodel xmlns="http://www.cisco.com/esc/esc">
  <flavors>
    <flavor>
      <name>testfl6</name>
      <vcpus>1</vcpus>
      <memory_mb>2048</memory_mb>
      <root_disk_mb>10240</root_disk_mb>
      <ephemeral_disk_mb>0</ephemeral_disk_mb>
      <swap_disk_mb>0</swap_disk_mb>
      <properties>
        <property>
          <name>hw:mem_page_size</name><value>2048</value>
        </property>
      </properties>
    </flavor>
```

```
  </flavors>
</esc_datamodel>
```

# PCI パススルーの設定

Intel VT-d 拡張機能は、物理デバイスをゲストに直接割り当てるためのハードウェアサポートを提供します。これにより、仮想デバイスは、複数のスイッチまたはブリッジを通過して物理インターフェイスに到達する際に伴うスループットの損失とパケット転送容量の減少を回避できます。

VT-d 拡張機能は、Red Hat Enterprise Linux を使用した PCI パススルーに必要です。拡張機能は、BIOS で有効にする必要があります。システム メーカーによっては、これらの拡張機能がデフォルトで無効になっている場合があります。

この手順では、BIOS の観点から VT-d を有効化する方法については説明しません。VT-d を有効にするには、サーバ製造時の BIOS 設定ガイドを参照してください。Linux カーネルの観点から、「intel_iommu = on」を grub 設定に追加することで、VT-d が有効になります。

### 始める前に

PCI パススルーをサポートするには、Intel チップセットで VT-d を有効にする必要があります。

**ステップ 1** Red hat Enterprise Linux で VT-d サポートを有効にします。

a) 「/etc/default/grub」を編集し、行の末尾に「intel_iommu＝on」を追加します。GRUB_CMDLINE_LINUX

**例**：

```
GRUB_TIMEOUT=5
GRUB_DEFAULT=saved
GRUB_DISABLE_SUBMENU=true
GRUB_TERMINAL_OUTPUT="console"
GRUB_CMDLINE_LINUX="rd.lvm.lv=rhel/swap crashkernel=auto rd.lvm.lv=rhel/root rhgb quiet
intel_iommu=on"
GRUB_DISABLE_RECOVERY="true"
```

b) grub2-mkconfig コマンドを実行して、grub.conf を再生成し、サーバを再起動します。

**例**：

```
grub2-mkconfig -o /boot/grub2/grub.cfg on BIOS systems
```

または

```
grub2-mkconfig -o /boot/efi/EFI/redhat/grub.cfg on UEFI systems
```

c) コマンドを実行して、IOMMU がアクティブになっていることを確認します。**dmesg | grep -iE "dmar|iommu"**

**例**：

IOMMU が有効になっている **dmesg** コマンドの出力例：

```
[ 0.000000] Kernel command line: BOOT_IMAGE=/vmlinuz-3.10.0-229.el7.x86_64
root=/dev/mapper/rhel-root
ro rd.lvm.lv=rhel/swap crashkernel=auto rd.lvm.lv=rhel/root rhgb quiet intel_iommu=on
[ 0.000000] Intel-IOMMU: enabled
```

**ステップ 2** Linux カーネルからデバイスをアンバインドします。

a) PCI パススルーが機能するには、デバイスが Linux カーネルドライバからアンバインドされている必要があります。これを実現するには pci_stub モジュールを使用します。

**例：**

```
Load pci_stub module "modprobe pci_stub"
```

b) PCI パススルーに使用するネットワークアダプタを見つけます。**lspci** を実行し、目的のネットワークカードの PCI アドレスをメモします。

**例：**

この例では、PCI パススルーに PCI デバイス 15: 00.0 を使用する必要があります。

```
12:00.0 Ethernet controller: Cisco Systems Inc VIC Ethernet NIC (rev a2)
13:00.0 Ethernet controller: Cisco Systems Inc VIC Ethernet NIC (rev a2)
14:00.0 Ethernet controller: Cisco Systems Inc VIC Ethernet NIC (rev a2)
15:00.0 Ethernet controller: Cisco Systems Inc VIC Ethernet NIC (rev a2)
```

c) ベンダー ID とデバイス ID を特定します。**lspci -n**の実行

**例：**

この部分的な出力では、ベンダー ID 1137 とデバイス ID 0071 が識別されます。

```
11:00.0 0c04: 1137:0071 (rev a2)
12:00.0 0200: 1137:0071 (rev a2)
13:00.0 0200: 1137:0071 (rev a2)
14:00.0 0200: 1137:0071 (rev a2)
15:00.0 0200: 1137:0071 (rev a2)
```

d) この設定を使用して、目的のデバイスを Linux カーネルドライバからアンバインドします。

強調表示されたテキストは、デバイス情報に合わせて変更する必要があります。

**例：**

```
echo "1137 0071" > /sys/bus/pci/drivers/pci-stub/new_id
echo 0000:15:00.0 > /sys/bus/pci/devices/0000:15:00.0/driver/unbind
echo 0000:15:00.0 > /sys/bus/pci/drivers/pci-stub/bind
```

e) **dmesg | grep stub** を実行して、これらのコマンドが成功したことを確認します。

**例：**

```
[ 276.705315] pci-stub 0000:15:00.0: claimed by stub
```

f) 変更を永続的にするには、pci-stub.ids を grub CMDLINE に追加し、grub を更新して、ホストを再起動します。

（注） このコードは、指定されたベンダー/デバイス ID を持つすべての vNIC（この例では 1137:0071）に適用されます。

例：

```
edit /etc/default/grub
GRUB_CMDLINE_LINUX="..pci-stub.ids=1137:0071"
grub2-mkconfig -o /boot/grub2/grub.cfg
reboot
```

**ステップ 3** OpenStack で nova を設定します。

a) （VendorID、productID）の組み合わせを使用して、（vendorID、productid）の組み合わせを使用した PCI デバイスの通過を許可するか、または通過を許可するデバイスの PCI アドレスを指定することにより、PCI パススルーに使用できる PCI デバイスを指定します。

```
pci_passthrough_whitelist={"vendor_id": "1137", "product_id": "0071"}
```

または

```
pci_passthrough_whitelist = [ {"address": "01:00.1"}, {"address": "02:00.1"} ]
```

b) PCI エイリアスと（製品 ID、ベンダー ID）の組み合わせマッピングを指定します。この設定では現在 PCI アドレスがサポートされていないため、ホワイトリストの PCI デバイスはすべて同じ名前になっています。

例：

```
pci_alias={"vendor_id":"1137", "product_id":"0071", "name":"nic1"}
```

c) nova.conf に次の追加の変更を加えます。

例：

```
scheduler_driver=nova.scheduler.filter_scheduler.FilterScheduler
scheduler_available_filters=nova.scheduler.filters.all_filters
scheduler_available_filters=nova.scheduler.filters.pci_passthrough_filter.PciPassthroughFilter
scheduler_default_filters=RamFilter,ComputeFilter,AvailabilityZoneFilter,ComputeCapabilitiesFilter,ImagePropertiesFilter
```

d) nova を再起動します。

例：

```
openstack-service restart nova
```

**ステップ 4** 属性 pci_passthrough を使用してフレーバーを作成します。エイリアスは <PCI_DEVICE_ALIAS>:<NUM_DEVICES_REQUESTED> に設定されています。PCI_DEVICE_ALIAS は、/etc/nova/nova.conf の pci_alias 設定から値を参照します。

例：

```
$ cat fl.xml
<?xml version='1.0' encoding='ASCII'?>
<esc_datamodel xmlns="http://www.cisco.com/esc/esc">
    <flavors>
        <flavor>
            <name>testfl6</name>
              <vcpus>1</vcpus>
              <memory_mb>2048</memory_mb>
```

```
                <root_disk_mb>10240</root_disk_mb>
                <ephemeral_disk_mb>0</ephemeral_disk_mb>
                <swap_disk_mb>0</swap_disk_mb>
                <properties><property>
                    <name>pci_passthrough:alias</name><value>nic1g:1</value>
                </property></properties>
            </flavor>
        </flavors>
</esc_datamodel>
$ sudo /opt/cisco/esc/esc-confd/esc-cli/esc_nc_cli edit-config ./fl.xml
```

# OpenStack で Heat オーケストレーション テンプレート（HOT）を使用した VPC-DI のオンボーディング

VPC-DI は、OpenStack 環境で仮想ネットワーク機能（VNF）として展開できます。VPC-DI は、仮想マシンの集合として実行され、VM には、ストレージ、ネットワーキング、および設定に関する固有の要件があります。OpenStack 環境では、オーケストレータは VPC-DI VM を起動するために必要なオブジェクトの作成を担当します。オーケストレータは、VM とそれらに関連付けられたオブジェクトを作成して終了する役割も担います。オーケストレータは OpenStack サービスを使用して、OpenStack にそのようなエンティティを作成します。

OpenStack は、テンプレートを介した VNF のネットワーク、コンピューティング、およびストレージトポロジを定義する、HEAT オーケストレーション テンプレート（HOT）と呼ばれるサービスを提供します。HEAT テンプレートは、VNF のインスタンスを展開するためのブループリントとして使用できます。

テンプレートの形式と ENV パラメータファイルの例については、この項の後半で説明します。

**ステップ 1** CF および SF の VPC-DI インスタンスの qcow イメージを取得します。

イメージを含む tarball ファイルには、リリース番号に応じて production.xxxxx.qvpc-di.qcow2.tgz というような名前が付けられています。tarball ファイルが開いている場合は、CF と SF の 2 つのイメージ（qvpc-di-cf.qcow2 と qvpc-di-xf.qcow2）が存在する必要があります。

**ステップ 2** **glance image-create** コマンドを使用するとすべての VPC-DI イメージが瞬時に作成されます。

**例**：

```
$ glance image-create --file qvpc-di-cf.qcow2 --container-format bare --disk-format
    qcow2 --is-public true --name cisco-qvpc-cf

$ glance image-create --file qvpc-di-sf.qcow2 --container-format bare --disk-format
    qcow2 --is-public true --name cisco-qvpc-sf
```

**ステップ 3** VPC-DI サンプルの初期化 tarball（vpc_HOT_sample.tgz）を取得します。

**ステップ 4** VPC-DI サンプルの初期化 tarball をローカルマシンにコピーします。

**ステップ 5** VPC-DI サンプルの初期化 tarball を任意のディレクトリに解凍します。拡張子が **.yml** と **.env** の 2 つのファイルがあります。

**ステップ 6** OpenStack の展開に応じて、ENV ファイルを編集します。

ネットワーク、可用性ゾーンなどの値を入力します。

ローカルディレクトリを参照して、テンプレートの送信元の **.yml** ファイルと環境の送信元の **.env** ファイルをクリックします。

**ステップ 7** 次のいずれかを実行します。

- [Project] > [Orchestration] > [Stacks] > [Launch Stack] に移動し、OpenStack ダッシュボードを使用して VPC-DI を展開します。
- CLI を使用して、**heat stack-create -f di.yml -e di.env** コマンドで HEAT を使用して VPC-DI を展開します。

**ステップ 8** [Status] フィールドが [Complete] であることを確認します。これはエラーがあることを示しています。

**ステップ 9** VPC-DI が統合されるまで待ちます。

**ステップ 10** VPC-DI 展開を削除するには、スタック名の横のチェックボックスをオンにし、[Delete Stack] をクリックします。

# VPC-DI Heat オーケストレーション テンプレート

この項では、Heat テンプレートの形式について説明します。VPC-DI HOT バージョンは 2013-05-23 です。テンプレートには、パラメータグループ、パラメータ、リソース、および出力の 4 つのセクションがあります。

## VPC-DI HOT パラメータ グループ

[parameter_groups] セクションでは、入力パラメータをグループ化する方法、およびパラメータを提供する順序を指定できます。これらのグループは、ダウンストリームユーザインターフェイスで予想される動作を説明するために使用されます。

**表 *6 : HOT* パラメータ**

| テンプレートでのパラメータ定義 | 注記 |
|---|---|
| `- label: images`<br>`  description: CF and SF images in qvpc-di`<br>`  parameters:`<br>`  - qvpc_image_cf`<br>`  - qvpc_image_sf` | テンプレートで定義されるイメージのリスト |

| テンプレートでのパラメータ定義 | 注記 |
| --- | --- |
| `- label: networks`<br>`  description: network configuration for DI`<br>`  parameters:`<br>`  - network_di_mgmt`<br>`  - network_di_internal`<br>`  - network_public`<br>`  - network_service1`<br>`  - network_service2`<br>`  - network_service3`<br>`  - network_service4` | テンプレートで定義されるネットワークのリスト<br>（注） この例では、4 個の SF サービスポートをリストしています。最大 12 個の SF サービスポートを定義できます。 |

## VPC-DI HOT パラメータ

Heat テンプレートは、ENV ファイル内に値を指定する必要があるパラメータの数を定義します。これらのパラメータについては、ここで説明します。各パラメータ定義は、heat テンプレートの **parameters** セクションに記載されています。サンプルの ENV ファイルは、パラメータの記述に従っています。

表 *7: HOT* パラメータ

| テンプレートでのパラメータ定義 | 注記 |
| --- | --- |
| `flavor_cf:`<br>`  type: string`<br>`  description: Flavor for Control Function VM`<br>`  default: m1.large` | CF を作成するために使用されるフレーバの名前。これは、5 つのデフォルトのフレーバのいずれかか、または OpenStack で定義されているカスタムフレーバになります。 |
| `flavor_sf:`<br>`  type: string`<br>`  description: Flavor for Service Function VM`<br>`  default: m1.large` | SF を作成するために使用するフレーバの名前。 |
| `availability_zone:`<br>`  type: string`<br>`  description: Availability_zone where the VNF`<br>`  should be created`<br>`  default: nova` | VNF が作成されている OpenStack の場所。 |
| `qvpc_image_cf:`<br>`  type: string`<br>`  label: Active CF image file in glance`<br>`  description: Active CF image ID or file in glance`<br>`  default: qvpc-di-<version>-cf.qcow2`<br>`  constraints:`<br>`  - custom_constraint: glance.image` | CF の VPC-DI イメージファイルの名前。このファイルは一瞬でアップロードされているはずです。 |

| テンプレートでのパラメータ定義 | 注記 |
|---|---|
| qvpc_image_xf:<br>  type: string<br>  label: SF image file in glance<br>  description: SF image ID or file in glance<br>  default: qvpc-di-<*version*>-xf.qcow2<br>  constraints:<br>  - custom_constraint: glance.image | サービス機能 VM の VPC-DI イメージファイルの名前。このファイルは一瞬でアップロードされているはずです。 |

| テンプレートでのパラメータ定義 | 注記 |
|---|---|
| network_public:<br>  type: string<br>  description: Network ID or Network Name of external network<br>  default: public<br>  constraints:<br>  - custom_constraint: neutron.network | 外部ネットワークのネットワーク ID または名前 |
| network_cf_mgmt:<br>  type: string<br>  description: Management Network ID or Name<br>  default: private<br>  constraints:<br>  - custom_constraint: neutron.network | VPC-DI 管理ネットワークの名前または識別子 。 |
| network_di_internal:<br>  type: string<br>  description: Unique QVPC-DI internal Network associated with this VNF<br>  default: private<br>  constraints:<br>  - custom_constraint: neutron.network | DI 内部ネットワークの名前または識別子。これは、VPC-DI で VM を相互接続するプライベート L2 ネットワークです。 |
| network_service#:<br>  type: string<br>  description: Network ID or Network Name of network to use for SF service ports<br>  default: cflocal<br>  constraints:<br>  - custom_constraint: neutron.network | サービスポートの名前または識別子。SF ごとに 1 ～ 12 個のサービスポートを定義できます。ここで、# はこの数値を表します。各サービスポートは異なるサービスを実行できます。 |
| network_core:<br>  type: string<br>  description: core network for keepalives<br>  default: core<br>  constraints:<br>  - custom_constraint: neutron.network | キープアライブメッセージに使用されるコアネットワークの名前または識別子。 |
| qvpc_vip_addr:<br>  type: string<br>  description: OAM IP Address shared between CF01 and CF02<br>  default: <*value*><br>  constraints:<br>  - custom_constraint: ip_addr | CF 間で使用される仮想 IP アドレス。 |

| テンプレートでのパラメータ定義 | 注記 |
|---|---|
| qvpc_vip_gateway:<br>type: string<br>description: IP Address of Default Gateway for OAM Network<br>default: <*value*><br>constraints:<br>- custom_constraint: ip_addr | 管理ネットワークのデフォルトゲートウェイ。 |
| vnf_name:<br>type: string<br>description: Unique name for this VNF instance<br>default: qvpc_di | この VNF インスタンスの一意の名前。この名前は、VNF の識別に使用されます。 |
| vnf_id:<br>type: string<br>description: Unique ID for this VNF instance<br>default: 0 | VNF インスタンスの ID。 |
| admin_password:<br>type: string<br>description: Default Administrator password for DI Access<br>default: Cisco123 | |
| snmp_community:<br>type: string<br>description: READ SNMP string for this VPC instance<br>default: public | |
| timezone:<br>type: string<br>description: TimeZone for this VF instance<br>default: us-pacific | |
| cf_domain_name:<br>type: string<br>description: Domain for this VF instance<br>default: localdomain | |
| az_cf<#>:<br>type: string<br>description: CF availability zone<br>default: <*value*> | 2 つの CF インスタンスそれぞれの可用性ゾーン。 |
| az_sf<#>:<br>type: string<br>description: CF availability zone<br>default:<*value*> | 各 SF インスタンスの可用性ゾーン。 |

これらのパラメータはそれぞれ、ENV ファイルを使用して VNF インスタンス用に定義されています。次に、ENV ファイルの例を示します。

```
parameters:
  # flavor defined for CF and SF in AIC
  flavor_cf: vsaegw_cf
  flavor_sf: vsaegw_sf

  # availability zone where the VNF instance should be deployed
  availability_zone: avzone-kvm-az01
```

```
  # vPC-DI glance images in qcow2
  qvpc_image_cf01: QVPCCF
  qvpc_image_sf: QVPCSF

  # Neutron Networks attached to vSAEGW instancenetwork_di_mgmt: oam_protected_net
  network_di_internal: saegw_di_internal_active_net
  network_service1: saegw_gn_net
  network_service2: saegw_sgi_net
  network_service3: saegw_support_net
  network_service4: saegw_icsr_li_net

  # VNF Instance Name
  vnf_name: qvpcDI_vsaegw

  # VNF Instance ID
  vnf_id: 01

  # Administrator user password
  admin_password: cisco123

parameters:
  flavor_cf:
    type: string
    description: Flavor for Control Function VM
    default: cisco-qvpc-cf
  flavor_sf:
    type: string
    description: Flavor for Service Function VM
    default: cisco-qvpc-xf
  qvpc_image_cf:
    type: string
    label: CF image file in glance
    description: CF image ID or file in glance
    default: qvpc-di-68031-cf.qcow2
    constraints:
      - custom_constraint: glance.image
  qvpc_image_sf:
    type: string
    label: SF image file in glance
    description: SF image ID or file in glance
    default: qvpc-di-68031-xf.qcow2
    constraints:
      - custom_constraint: glance.image
  network_public:
    type: string
    description: Network ID or Network Name of external network
    default: public
    constraints:
      - custom_constraint: neutron.network
  network_cf_mgmt:
    type: string
    description: Management Network ID or Name
    default: cf-mgmt
    constraints:
      - custom_constraint: neutron.network
  network_di_internal:
    type: string
    description: Unique QVPC-DI internal Network associated with this VNF
    default: di-internal
    constraints:
      - custom_constraint: neutron.network
  network_service1:
    type: string
    description: Transport Interface (Gn/S11/S1-u/S5) in to SAEGW Context
    default: service1
```

```
    constraints:
      - custom_constraint: neutron.network
  network_service2:
    type: string
    description: Transport Interface (Data, Voice, LI VLANs) in SGi Context
    default: service2
    constraints:
      - custom_constraint: neutron.network
   network_core:
    type: string
    description: core network for keepalives
    default: core
    constraints:
      - custom_constraint: neutron.network

# vip_addr and vip_gateway are automatically retrieved from the management network
  qvpc_vip_addr:
    type: string
    description: OAM IP Address shared between CF01 and CF02
    default: 172.16.181.2
    constraints:
      - custom_constraint: ip_addr
  qvpc_vip_gateway:
   type: string
   description: IP Address of Default Gateway for OAM Network
   default: 172.16.181.1
   constraints:
      - custom_constraint: ip_addr
  vnf_name:
    type: string
    description: Unique name for this VNF instance
    default: qvpc_di
  vnf_id:
    type: string
    description: Unique ID for this VNF instance
    default: 0
  admin_password:
    type: string
    description: Default Administrator password for DI Access
    default: Cisco123
  snmp_community:
    type: string
    description: READ SNMP string for this VPC instance
    default: public
  timezone:
    type: string
    description: TimeZone for this VF instance
    default: us-pacific
  cf_domain_name:
    type: string
    description: Domain for this VF instance
    default: localdomain
  az_cf1:
    type: string
    description: CF availability zone
    default: conway1
  az_cf2:
    type: string
    description: CF availability zone
    default: conway2
  az_sf3:
    type: string
    description: SF3 availability zone
    default: conway3
```

```
  az_sf4:
    type: string
    description: SF6 availability zone
    default: conway4
```

# VPC-DI HOT リソース

テンプレートのリソースセクションでは、制御機能（CF）およびサービス機能（SF）の VM と、それらのポートをそれぞれ定義します。

### 管理ネットワーク

```
# Create port on management network and reserve a virtual IP address
qvpc_vip_port:
  type: OS::Neutron::Port
  properties:
      network: {get_param: network_di_mgmt}
      fixed_ips:
        - subnet_id: {get_param: subnet_id_di_mgmt}

# Associate a floating IP address to the virutal port
qvpc_vip_floating_ip:
  type: OS::Neutron::FloatingIP
  properties:
      floating_network: {get_param: network_public}
      port_id: {get_resource: qvpc_vip_port}
```

VIP ポートは、VPC-DI へのアクセスに使用される仮想 IP ポートです。VIP ポートの IP アドレスは、Day 0 設定で設定できます。

## CF の HOT リソース

Heat テンプレートでは、VNF を使用する 2 つの CF VM それぞれを定義する必要があります。この定義では、DI 内部ネットワークに接続するポートと CF 管理ネットワークに接続するポートを設定し、StarOS 起動パラメータファイルと StarOS Day 0 設定ファイルを指定します。次に最初の CF の定義と説明を示します。2 番目の CF も同様に定義されます。

### CF DI 内部ネットワーク

このセクションでは、CF DI 内部ネットワークを作成します。設定する必要がある 2 つの CF のそれぞれに対して、このセクションを 2 回使用します。# は 1 または 2 のいずれかです。

```
# Port connected to unique DI-network
qvpc_cf_0#_port_int:
  type: OS::Neutron::Port
  properties:
    network: {get_param: network_di_internal}
    allowed_address_pairs:
      -ip_address: "172.16.0.0/18"
```

**qvpc_cf_#_port_int** は DI 内部ネットワークに接続されているポートです。ネットワークの値は、ENV ファイルから取得したパラメータ **network_di_internal** から抽出されます。

プロパティの allowed_address_pairs は、各 DI 内部ポートに存在する必要があります。di_internal ポートには、17.16.0.0/18 内の VPC-DI によって Neutron 内のアドレスとは異なる IP アドレスが割り当てられるため、allowed_address_pairs プロパティを設定してそれらのアドレスのすべてのトラフィックがそのポートを通過できるようにします。許可されたアドレスペアの拡張機能はポート属性を拡張し、ネットワークに関連付けられているサブネットに関係なく、ポートを通過できる任意の MAC アドレスまたは IP アドレス（CIDR）のペアを指定できるようにします。

### CF 管理ネットワーク

このセクションでは、CF 管理ネットワークを作成します。設定する必要がある 2 つの CF のそれぞれに対して、このセクションを 2 回使用します。# は 1 または 2 のいずれかです。

```
# Port connected to the management network
qvpc_cf_0#_port_mgmt:
  type: OS::Neutron::Port
  properties:
    network: {get_param: network_di_mgmt}
    allowed_address_pairs:
      - ip_address: {get_param: qvpc_vip_addr}
```

**qvpc_cf_#_port_mgmt** は、OAM ネットワークに接続されているポートのポート定義を表します。値は、ENV ファイルから取得したパラメータ **network_di_mgmt** から抽出されます。

### SSH キー

DI VM 間通信は、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。これらのキーは、Heat 展開の一部として渡されます。公開キーと秘密キーが必要です。

公開 SSH キーと秘密 SSH キーを生成します。公開キーを含む *user_key.pub* という名前のファイルを作成します。秘密キーを含む *user_key* という名前のファイルを作成します。これらのファイルの両方が設定ドライブに保存されていることを確認します。これらのファイルは、Heat によって参照されます。

```
      personality:
        "user_key.pub": |
                    ssh-rsa
<public_key>
        "user_key": |
                    -----BEGIN RSA PRIVATE KEY-----
<private_key>
                    -----END RSA PRIVATE KEY-----
```

### CF VM の作成

このセクションでは、CF VM を作成します。このセクションは作成する必要がある 2 つの CF それぞれに 1 回ずつ、2 回使用します。# は 1 または 2 のいずれかです。

```
qvpc_cf_0#:
  type: OS::Nova::Server
  properties:
    # Create VM of format "<vnf_name>_cf_0#"
```

```
name:
  str_replace:
    template: ${VF_NAME}_cf_0#
    params:
      ${VF_NAME}: {get_param: vnf_name}
# Use active CF image and CF Flavor
image: {get_param: qvpc_image_cf1 }
flavor: {get_param: flavor_cf }
networks:
  - port: {get_resource: qvpc_cf_0#_port_int}
  - port: {get_resource: qvpc_cf_0#_port_mgmt}
config_drive: True
```

CF VM（**qvpc_cf_#**）は、以前に定義したパラメータで作成され、「<vnf_name>_cf_#」の表記法に従って名前が付けられます。**vnf_name** は、VNF の作成に使用されるイメージとフレーバと同様に、ENV ファイルから取得されます。

### StarOS の Day 0 設定

ここで指定する Day 0 設定は、DI インターフェイス、システムホスト名を設定し、*personality* プロパティを使用して SSH アクセスと SFTP アクセスを有効にします。

```
# Metadata to provide cloud-init capability to VPC-DI
  personality:
    "staros_param.cfg":
        str_replace:
            template: |
                CARDSLOT=$CARD_NUMBER
                CARDTYPE=$CARD_TYPE
                CPUID=$UUID
                DI_INTERFACE_MTU=1500
                DI_INTERFACE=TYPE:virtio_net-1
                MGMT_INTERFACE=TYPE:virtio_net-2
                VNFM_INTERFACE=TYPE:virtio_net-3
                VNFM_IPV4_ENABLE=true
                VNFM_IPV4_DHCP_ENABLE=true
                VNFM_PROXY_ADDRS=192.168.180.92,192.168.180.91,192.168.180.93
            params:
                $CARD_NUMBER: 1
                $CARD_TYPE: "0x40030100"
                $UUID: 0
    "staros_config.txt":
        str_replace:
            template: |
                config
                    system hostname $VF_NAME-cf-$CARD_NUMBER
                    clock timezone $TIMEZONE
                    ssh key-gen wait-time 0
                    context local
                        administrator admin password $ADMIN_PASS ftp
                            interface LOCAL1
                                ip address $CF_VIP_ADDR 255.255.255.0
                        #exit
                        ip route 0.0.0.0 0.0.0.0 $CF_VIP_GATEWAY LOCAL1
                        ip domain-lookup
                        ip domain-name $CF_DOMAIN_NAME
                        ip name-servers $CF_VIP_GATEWAY
                        ssh generate key
                        server sshd
                            subsystem sftp
```

```
                            #exit
                            server confd
                                confd-user admin
                        #exit
                        port ethernet 1/1
                            bind interface LOCAL1 local
                            no shutdown
                        #exit
                        snmp community $SNMP_COMMUNITY read-only
                    end
                params:
                    $CARD_NUMBER: 1
                    $VF_NAME: {get_param: vnf_name}
                    $TIMEZONE: {get_param: timezone}
                    $ADMIN_PASS: {get_param: admin_password}
                    $SNMP_COMMUNITY: {get_param: snmp_community}
                    $CF_DOMAIN_NAME: {get_param: cf_domain_name}
                    $SLOT_CARD_NUMBER: 1
                   #$CF_VIP_ADDR: {get_attr: [qvpc_vip_port, fixed_ips, 0, ip_address]}

                    $CF_VIP_ADDR: 172.16.181.2
                  #$CF_VIP_GATEWAY: { get_attr: [qvpc_vip_port, subnets, 0, gateway_ip]
 }
                    $CF_VIP_GATEWAY: 172.16.181.1
        "user_key.pub": |
                    ssh-rsa
<public_key>
        "user_key": |
                    -----BEGIN RSA PRIVATE KEY-----
<private_key>
                    -----END RSA PRIVATE KEY-----
```

$CARD_NUMBER はスロットの番号を参照します。ここでは 1 ですが、2番目の CF では 2 です。

## SF の HOT リソース

VPC-DI に展開する各サービス機能（SF）VM を定義するには、Heat テンプレートを使用します。SF ごとに、DI 内部ネットワークと、SF に必要なサービスポートのそれぞれに接続するようにポートを設定する必要があります。設定できるポートは最大 12 個です。この例では、4 つのサービスポートがある SAE ゲートウェイに使用する 1 つの SF を作成します。必要に応じて、SF ごとに同様の設定を繰り返す必要があります。

### SF でのポートの定義

```
# Create port for DI-Internal Network
  qvpc_sf_03_port_int:
    type: OS::Neutron::Port
    properties:
        network: {get_param: network_di_internal}
        allowed_address_pairs:
          - ip_address: "172.16.0.0/18"
```

**qvpc_sf_#_port_int**は、内部 DI ネットワークに接続するポートです。# は SF の番号であり、3 から SF に許可されている最大数までの番号です。ネットワークの値は、ENV ファイルから取得したパラメータ **network_di_internal** から抽出されます。

```
# Create first service port (document as per your use)
  qvpc_sf_03_port_svc_01:
    type: OS::Neutron::Port
    properties:
      network: {get_param: network_service1}
```

**qvpc_sf_#_port_svc_01**は、最初のサービスポートです。ポートには、1 から 12 までの連続番号が付けられます。ネットワークの値は、ENV ファイルから取得したパラメータ **network_service1** から抽出されます。

```
# Create second service port (document as per your use)
  qvpc_sf_03_port_svc_02:
    type: OS::Neutron::Port
    properties:
      network: {get_param: network_service2}
      allowed_address_pairs:
        - ip_address: "192.168.10.0/24"
# Create third service port  (document as per your use)
  qvpc_sf_03_port_svc_03:
    type: OS::Neutron::Port
    properties:
      network: {get_param: network_service3}
# Create forth service port  (document as per your use)
  qvpc_sf_03_port_svc_04:
    type: OS::Neutron::Port
    properties:
      network: {get_param: network_service4}
```

残りの 3 つのサービスポートが作成されます。それぞれが ENV ファイルからネットワーク情報を取得します。必要に応じて追加のサービスポートを作成できます。

### SF VM の作成

```
qvpc_sf_03:
  type: OS::Nova::Server
  properties:
# Create VM name of format "<vnf_name>_sf_0<num>"
    name:
      str_replace:
          template: ${VF_NAME}_sf_03
          params:
            ${VF_NAME}: {get_param: vnf_name}
# Use SF image and SF Flavor
    image: { get_param: qvpc_image_sf }
    flavor: { get_param: flavor_sf }
    networks:
      - port: {get_resource: qvpc_sf_03_port_int}
      - port: {get_resource: qvpc_sf_03_port_svc_01}
      - port: {get_resource: qvpc_sf_03_port_svc_02}
      - port: {get_resource: qvpc_sf_03_port_svc_03}
      - port: {get_resource: qvpc_sf_03_port_svc_04}
    config_drive: True
```

SF **qvpc_sf_#** は「vnf_name_sf_0#」形式の名前で作成されます。ここで、vnf_name は ENV ファイルから取得した VNF 名の値で、# は SF のスロットです。サービスポートの値は、Heat テン

プレートに前もって定義されています。イメージとフレーバは、ENV ファイルからも取得されます。

各 SF はテンプレートと同様に定義されます。

### パーソナリティの設定

Day 0 と Day 1 の設定は、パーソナリティプロパティを使用して VNF に挿入されます。VPC-DI は、システムにパーソナリティプロパティを適用し、次に示すように、 Heat テンプレートからのこのメタデータを予期します。

パーソナリティは、起動パラメータファイルを定義します。起動パラメータの詳細については、起動パラメータの設定 （29 ページ）を参照してください。

```
# Associate VM to unique slot (>2) and identify that its a SF
       config_drive: True
      personality:
        "staros_param.cfg":
            str_replace:
                template: |
                    CARDSLOT=$CARD_NUMBER
                    CARDTYPE=$CARD_TYPE
                    CPUID=$UUID
                    DI_INTERFACE_MTU=1500
                params:
                    $CARD_NUMBER: 3
                    $CARD_TYPE: "0x42070100"
                    $UUID: 0
        "user_key.pub": |
                    ssh-rsa
<public_key>
        "user_key": |
                    -----BEGIN RSA PRIVATE KEY-----
                    <private_key>
                    -----END RSA PRIVATE KEY-----
```

DI VM 間通信は、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。これらのキーは、Heat 展開の一部として渡されます。公開キーと秘密キーが必要です。

公開 SSH キーと秘密 SSH キーを生成します。公開キーを含む *user_key.pub* という名前のファイルを作成します。秘密キーを含む *user_key* という名前のファイルを作成します。これらのファイルの両方が設定ドライブに保存されていることを確認します。これらのファイルは、上記のように Heat によって参照されます。

## VPC-DI HOT 出力

heeatテンプレートの [outputs] セクションには、テンプレートを使用した出力が定義されます。出力を表示するには、**[Project] > [Orchestration] > [Stacks]** に進み、展開した heat スタックを選択します。[Overview] タブには、heat スタックからの出力が表示されます。

コマンドラインで **heat stack-show $[stack_name]** コマンドを実行して、heat スタックからの出力を表示することもできます。

VPC-DI に定義される可能性のある出力のタイプの例を次に示します。

```
qvpc_floating_ip:
  description: Floating IP of qvpc-di VIP
  value: { get_attr: [qvpc_vip_floating_ip, floating_ip_address]}
CF1_networks:
  description: The networks of the deployed CF-1
  value: { get_attr: [qvpc_cf_01, networks] }
CF2_networks_2:
  description: The networks of the deployed CF-2
  value: { get_attr: [qvpc_cf_02, networks] }
port_1_int:
  description: The port of the deployed server 1, di-internal
  value: { get_attr: [qvpc_cf_01_port_int, mac_address] }
port_1_mgmt:
  description: The port of the deployed server 1, cf-mgmt
  value: { get_attr: [qvpc_cf_01_port_mgmt, mac_address] }
port_2_int:
    description: The port of the deployed server 2, di-internal
    value: { get_attr: [qvpc_cf_02_port_int, mac_address] }
port_2_mgmt:
    description: The port of the deployed server 2, cf-mgmt
    value: { get_attr: [qvpc_cf_02_port_mgmt, mac_address] }
```

# VMware のインストールに関する注意事項

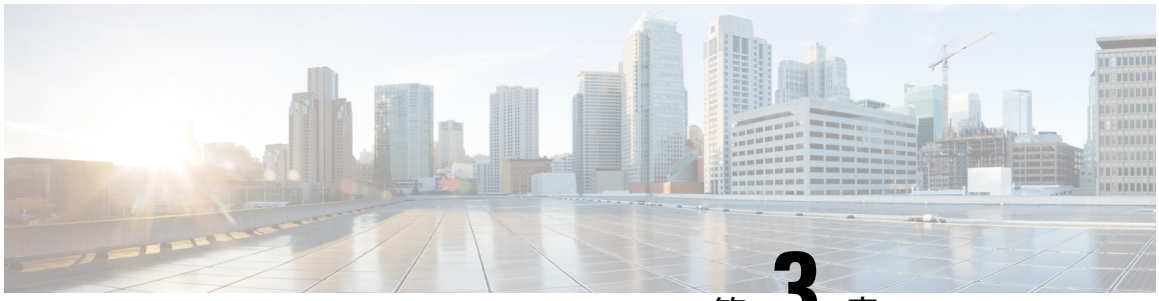
DI VM 間通信は、外部から提供された SSH キーによる認証によってのみ可能になりました。公開キーと秘密キーが必要です。これらのキーは、ISO の一部として VM を起動する前に指定する必要があります。

キーは外部ホストで生成され、その後 VM に接続する必要がある ISO 内にパッケージ化される必要があります。キーと ISO ファイルは、次のように生成されます。

```
$ mkdir iso
$ ssh-keygen -t rsa -N "" -C "root@localhost" -f iso/user_key
$ genisoimage -o vpcdi_keys.iso iso
```

ISO ファイルが生成されたら、VM の電源を投入し、CD-DVD ROM にマッピングします。これを行うには、vSphere 内でリストから VM（CF または SF）を選択し、上部の近くにあるオプションバーから CD/DVD アイコンをクリックします。次に、[Connect to ISO image on local disk] を選択し、ISO を選択します。すべての VM（CF と SF）に対してこの手順を繰り返します。

キーがマッピングされたら、適切な起動の優先順位を設定して VPC-DI をリロードすることによって VPC-DI の起動設定をイメージにポイントします。



第 3 章

# システムの運用と設定

シスコ VPC-DI には、モバイル パケット コア ネットワーク用のサブスクライ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

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、およびサブスクライバを設定した後、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。サブスクライバのドメイン名が、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、サブスク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、非ローカルの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

認証、許可、およびアカウンティング（AAA）サーバは、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ティングレコードを維持します。AAA サーバは、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）サーバとホーム AAA（HAAA）サーバ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 サーバはキャリアのネットワークに存在します。HAAA サーバは、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、アカウンティングデータは AAA プロキシサーバ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。

# サブスクライバ

サブスクライバは、サービスのエンドユーザです。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。

サブスクライバには、主に次の 3 つのタイプがあります。

- **RADIUS ベースのサブスクライバ**：最も一般的なタイプのサブスクライバであるこれらのユーザは、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名によって識別されます。これらは、RADIUS AAA サーバで設定され、認証されます。

  認証が成功すると、サブスクライ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）、およびサブスクライバが持っている権限などを決定します。

  **重要** RADIUS AAA サーバ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ライバの属性とパラメータよりも優先されます。

- **ローカルサブスクライバ**：これらは、主にテスト目的で使用されるサブスクライバであり、特定のコンテキスト内で設定および認証されます。RADIUS ベースのサブスクライバとは異なり、ローカルサブスクライバのユーザプロファイル（RADIUS ベースのサブスクライ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。

  ローカルサブスクライバのプロファイルが最初に作成されると、そのサブスクライバの属性はシステムのデフォルトに設定されます。すべてのサブスクライ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ライバが含まれます。ローカルプロファイルの属性を設定すると、サブスクライバごとに変更が行われます。

  **重要** ローカルサブスクライ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。

- **管理サブスクライバ**：管理ユーザは、CLI を通じてシステムをモニタ、制御、および設定できる権限を持つユーザです。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。管理ユーザは通常、ローカルコンテキスト内でローカルサブスクライ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。ローカルサブスクライバと同様に、管理者サブスクライバのユーザプロファイルは、サブスクライ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、管理サブスクライバは 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またはサブスクライ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（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* のユーザ名のシンタックスを使用する必要があります。

**重要** リリース 20.0 以降の信頼できる StarOS のビルドでは、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を認証するために AAA サーバと連携する場合に、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 サーバ上に設定するといった柔軟性を提供します。ローカルに設定されたユーザ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 サーバでユーザプロファイルを設定している場合、システムは AAA サーバ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 *5:* コンテキストレベルの管理ユーザの *AAA* コンテキスト

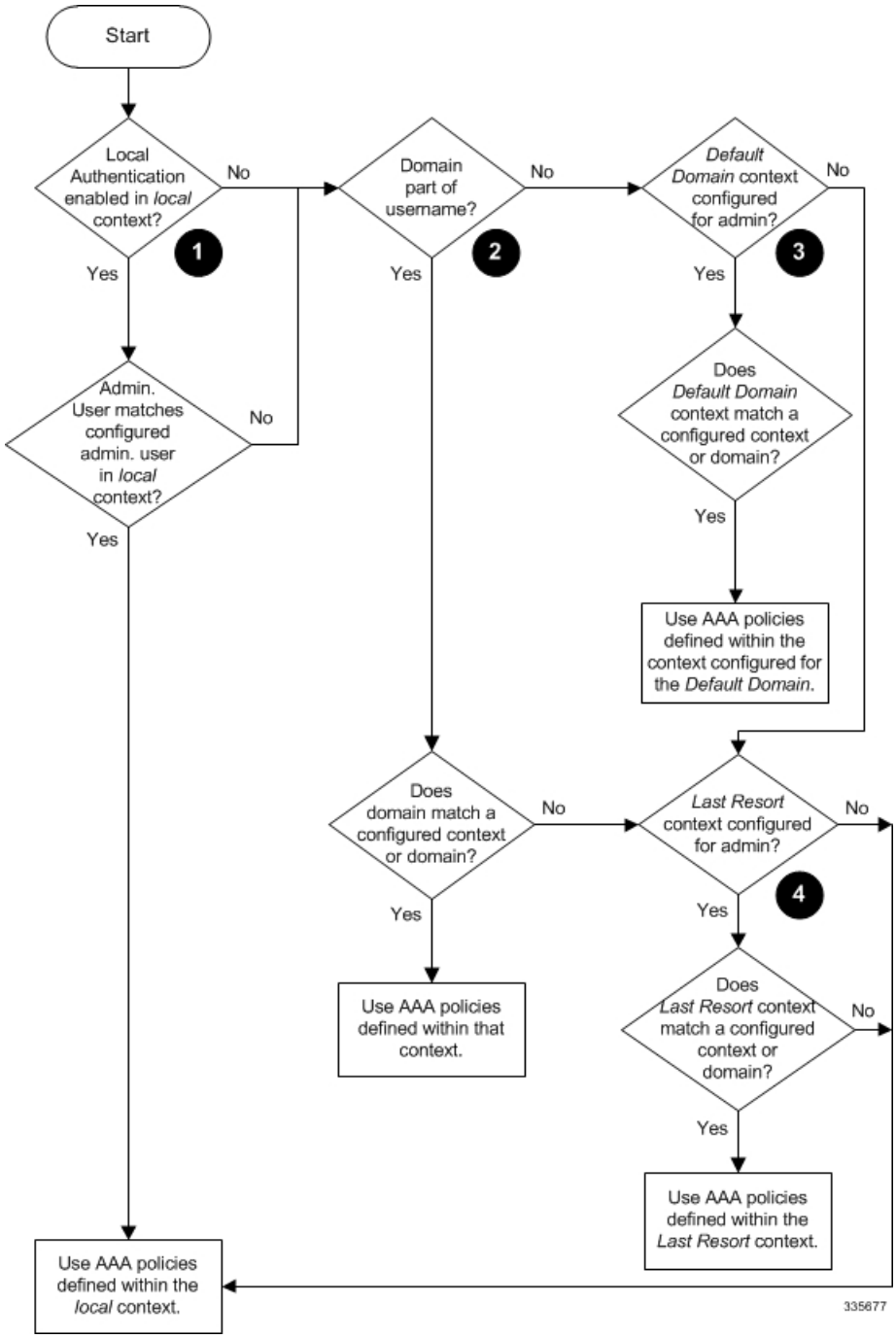


表 *8* : コンテキストレベルの管理ユーザの *AAA* コンテキストにおける選択

| 項目 | 説明 |
|---|---|
| 1 | 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がシステムによって判別されます。<br>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。<br>管理ユーザのユーザ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。 |
| 2 | システムでローカル認証が無効になっている場合、または管理ユーザのユーザ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。<br>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。<br>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。<br>ユーザ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。 |
| 3 | ユーザ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。<br>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。<br>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。 |

| 項目 | 説明 |
|---|---|
| 4 | ドメインがユーザ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。<br><br>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。<br><br>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。 |

リリース 21.4 以降（信頼できるビルドのみ）：

- ユーザは、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と、そのユーザは拒否されます。
- ユーザが「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_addrs**

## 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ようにするには、『*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 ファイルを適用しても、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**：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。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（:）で区切られたの 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）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ベ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 192.168.100.1/24 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 192.168.100.0、または同等のサブネットマスク 255.255.255.0 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、$2^{\text{address size - mask}}$ 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」文字を使用しないでください。

- **&**（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）

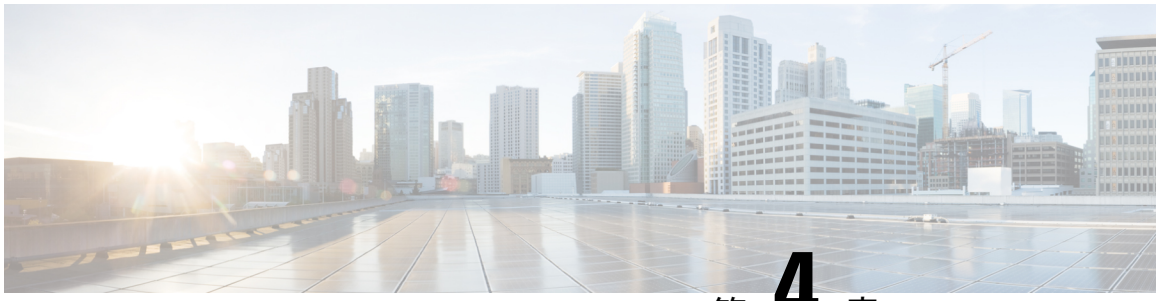
次の文字を使用して、グローバルAAA機能のユーザ名からドメインを区切ることができます。

- @（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 4 章

# 使用する前に



## StarOS の初期設定

すべての VM での VPC-DI のインストールが正常に完了したら、アクティブな制御機能（CF）VM を使用して StarOS パラメータのセットを設定する必要があります。その後で、VPC-DI インスタンス内の VM が再起動されるたびにアクセスされる、アクティブな CF 上の設定ファイルにこれらの設定を保存します。スタンバイ CF およびすべてのサービス機能（SF）VM は、この設定ファイルをアクティブ CF から読み取ります。

この章では、アクティブな CF コンソールポートに接続し、最初のローカルコンテキスト管理設定を作成する手順について説明します。

## 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
- SSHを介したリモート CLI アクセスのための VPC-DI インスタンスの設定

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、アクティブな CF VM のコンソールポートにログインします。

**ステップ 2** CLI プロンプトで、次のように入力します。

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

ローカルコンテキストは、 VPC-DI インスタンス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** コンテキストレベルのセキュリティ管理者を VPC-DI インスタンスに設定するには、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]cf_host_name(config-ctx) exit
[local]cf_host_name(config)
```

**ステップ 6** 次のコマンドを入力して、VPC-DI インスタンス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]cf_host_name(config) system hostname cf_host_name
```

*cf_host_name* は、VPC-DI インスタンスがネットワーク上で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-di」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]cf_host_name(config) context local
[local]cf_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]cf_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。StarOS 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 モードを開始すると、次のプロンプトが表示されます。

```
[local]cf_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]cf_host_name(config-if-eth) exit
[local]cf_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、VPC-DI インスタンス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]cf_host_name(config-ctx) exit
[local]cf_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。

```
[local]cf_host_name(config) port ethernet slot/port
```

VPC-DI の場合 、*slot* は仮想シャーシ内の CF または SF VM に対応します。ハイパーバイザは VPC-DI インスタンスの初期設定時に、各 VM に一意のスロット番号を割り当てます。スロット番号 1 と 2 は CF VM に割り当てられ、スロット番号 3 ～ 32 は SF VM に割り当てられます。CF はポート 1 のみをサポートします。各 SF は、1 ～ 4 番の 4 つの vNIC をサポートし、対応する仮想イーサネットポート 10 ～ 14 番を使用します。SF ポート番号 10 を設定する必要があります。

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]cf_host_name(config-port-slot/port) bind interface interface_name local
[local]cf_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]cf_host_name(config-port-slot/port) exit
[local]cf_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。

# システム管理ユーザの設定

この項では、セキュリティ管理者がユーザ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、ユーザがすでに使用されている機密ユーザ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+ サーバユーザ
- ローカルユーザのユーザ
- AAA コンテキストユーザ

TACACS+ サーバユーザ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。ローカルユーザのユーザと AAA コンテキストユーザ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、ユーザがユーザ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。

**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） リリース 4.0 以降では、**v1-rsa** キーワードは削除されており、 **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

SSHv2 RSA は、StarOS でサポートされる SSH の唯一のバージョンです。SSHv1 RSA および SSHv2 DSA で以前サポートされていたキーワードは、StarOS CLI 内で削除されたか、または隠されています。

重要 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。StarOS は、以前のリリースで作成された既存のスクリプトとコンフィギュレーション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は、解析時にエラーメッセージを生成します。

SSH プロトコルのバージョン 1 は、セキュリティの脆弱性が原因で廃止されました。**v1-rsa** 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** コマンドのために削除されました。SSHv1-RSA キーを使用するスクリプトまたは設定を実行すると、エラーメッセージが返され、イベントログが生成されます。次に、エラーメッセージの出力例を示します。

```
CLI print failure Failure: SSH V1 contains multiple structural vulnerabilities and is
no longer considered secure. Therefore we don't support v1-rsa SSH key any longer, please
 generate a new v2-rsa key to replace this old one.
```

**v1-rsa** キーを含む設定からシステムが起動する場合、SSH を介してログインするときに起動の失敗が予想されます。回避策は、コンソールポートを介してログインし、新しい ssh v2-rsa キーを再生成し、サーバ sshd を設定することです。その後、ssh を介してログインできるようになります。

コンテキスト コンフィギュレーション モードの **ssh** コマンドでは、**v2-dsa キーワード**が隠されるようになりました。

**v1-rsa** キーワードは、Exec モードの **show ssh key** CLI コマンドから削除されました。

# 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-DI）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。

```
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 | • ASR 5500<br>• VPC-DI<br>• 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 | • *ASR 5500 System Administration Guide*<br>• *Command Line Interface Reference*<br>• *VPC-DI* システム管理ガイド<br>• *VPC-SI System* アドミニストレーション 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 |
|---|---|
| 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**：

- *alogorithms*：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
```

## *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：

- *alogorithms*：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 サーバによって使用される公開キーと秘密キーのペアを生成します。**v1-rsa** キーワードが削除されており、 **v2-dsa** キーワードが **ssh generate** CLI コマンド内に隠されています。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** *20.0* よりも前のリリースでは、最大 64 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。リリース *20.0* 以降では、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 ユーザアクセス

ユーザが、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のみが、SSH を介してホストに接続できます。ユーザ名には、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、任意のユーザによる無制限のアクセスを許可します。

## 許可済みユーザリストの作成

**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アカウントの組み合わせを使用して SSH によるユーザログインの試行を認証します。

- ユーザは、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 ID が異なるユーザが同じ公開 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。ユーザの認証では、許可済みキーとユーザ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にプッシュします。

（注） デフォルトでは、StarOS は外部サーバへの 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 秘密キー情報とキータイプを入力します。

```
[local]host_name(config-ssh)# ssh key private_key_string length key_length [ type v2-rsa ]
[local]host_name(config-ssh)#
```

**key** *private_key_string* は、秘密キーの値を 1 ～ 4499 文字の英数字の文字列として指定します。

**length** *key_length* は、0 ～ 65535 の整数でキーの長さをバイト単位で指定します。

**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。

**ステップ 3** 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 ]
[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。

**ステップ 4** 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 5** 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)#
```

## 外部サーバへの SSH クライアント公開キーのプッシュ

このサーバへの 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、外部サーバで有効なユーザ名を 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 |
|---|---|---|
| **ステップ 1** | 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。 | `[local]host_name` **configure**<br>`[local]host_name(config)` |
| **ステップ 2** | 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。 | `[local]host_name(config)` **context local**<br>`[local]host-name(config-ctx)` |
| **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スロット番号とポート番号を入力します。 | `[local]host_name(config-ctx) 1/1`<br>`[local]host_name(config-if-eth)` |
| **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。 | `[local]host_name(config-if-eth)` **{ ip \| ipv } address** *ipaddress subnet_mask* `secondary` |
| **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]host_name(config-if-eth)` **end** |
| **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。 | `[local]host_name` **show config context local**<br>CLI 出力は次の例のようになります。<br>`config`<br>`  context local`<br>`  interface interface_name`<br>`  ip address ipaddress subnetmask`<br>`  ip address ipaddress subnetmask secondary`<br>`  exit` |
| **ステップ 7** | インターフェイスとポートの設定の確認と保存（105 ページ）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 | • ASR 5500<br>• VPC-DI<br>• VPC-SI |
| 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 | • *ASR 5500 System Administration Guide*<br>• *Command Line Interface Reference*<br>• *VPC-DI* システム管理ガイド<br>• *VPC-SI System* アドミニストレーション 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 |
|---|---|
| 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**

  **リリース 20.x ～ 21.15（通常のビルドのみ）**

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

- **MAC**

  **リリース 20.x ～ 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**

  **リリース 20.x ～ 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-cbc
  ```

  **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
```

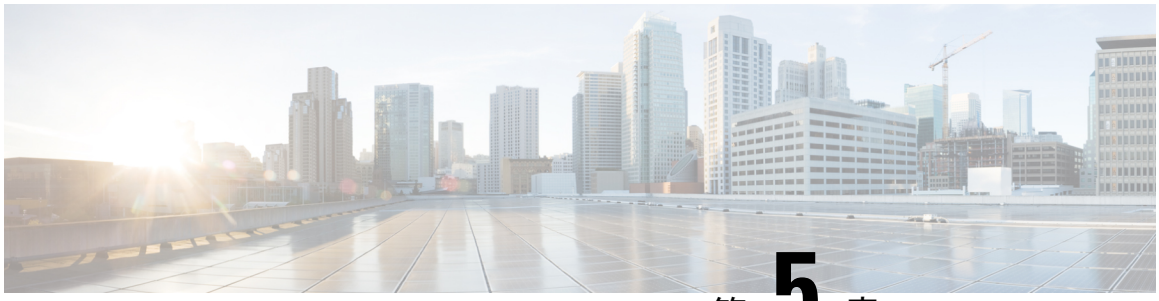
設定ディスクまたはローカルフラッシュ上の設定ファイルを表示するには、**show cloud configuration** *card_number* コマンドを使用します。フラッシュメモリ上のロケーション パラメータ ファイルは、インストール時に定義されます。また、ディスク構成は通常、オーケストレーションによって作成され、カードに接続されます。次に、カード 1 でのこのコマンドの出力例を示します。

```
[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 5 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** VPC-DI の展開の場合にこの章の手順を実行するには、アクティブな制御機能（CF）VM にログイン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、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

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-DI の場合、スロットは CF または SF VM に対応します。ハイパーバイザは VPC-DI インスタンスの初期設定時に、各 VM に一意のスロット番号を割り当てます。スロット番号 1 と 2 は CF VM に割り当てられ、スロット番号 3 ～ 32 は SF VM に割り当てられます。各 SF は 1 ～ 12 の番号が付いた 4 個の vNIC と、10 ～ 21 の番号が付いた対応する仮想イーサネットポートをサポートします。SF ポート番号 10 を設定する必要があります。

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 （アクティブ CF）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）のすべてのサーバでも設定する必要があります。StarOS の NTP 設定は、COTS サーバの NTP 設定と一致している必要があります。

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、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 サーバ用に NTP が設定されていること。サーバが 3 台以上ある場合は、サーバ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、サーバの数が多いほど（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 サーバの IP アドレスを入力します。

重要 少なくとも 3 台（できれば 4 台）の NTP サーバ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。

# ローカルソース使用した NTP サーバの設定

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中間 NTP サーバ上に設定されます。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。

# 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 サーバに関する情報が表示されます。2 つの NTP サーバ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、サーバの現在の状態を示します。<br>• ( ) Rejected/No response<br>• X False tick<br>• . Excess<br>• - Outlyer<br>• + Candidate<br>• # Selected<br>• * System peer<br>• (o) PPS peer |
| refid | サーバが同期している最後に報告された NTP の参照。 |
| st | NTP サーバ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 サーバ間の電話機の設定ユーティリティ。 |
| reach | この NTP サーバへの過去 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-DI 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-DI 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に確認のプロンプト（[Are you sure? [Yes|No]:]）が表示されます。

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（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 コマンドの次のいずれかのカテゴリを指定します。
  - card
  - clear
  - copy
  - debug
  - delete
  - 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、管理ユーザのタイプ（コンテキストレベルまたはローカルユーザ）に従って分類されます。

**重要** これらのユーザ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。

# ユーザ名の文字制限

ユーザ名には、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 ユーザ名とローカルユーザ名に適用されます。

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の設定

このユーザ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アカウント、または外部 RADIUS サーバまたは TACACS+ サーバの両方に当てはまります。これらのユーザ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。

**ローカルユーザのユーザ名**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、管理ユーザは、対応するユーザ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。

## 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、『*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 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と、そのユーザは拒否されます。
- ユーザが「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_addrs**

## コンテキストレベルの管理ユーザ設定の確認

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
```

# ローカルユーザの管理ユーザの設定

ローカルユーザタイプは、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** リリース 20.0 以降の信頼できる StarOS ビルドでは、ローカルユーザデータベースは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、ローカルユーザの管理ユーザを設定します。

```
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。

ローカルユーザデータベースの詳細については、ローカルユーザデータベースの更新とダウングレード （122 ページ）を参照してください。

## ローカルユーザ設定の確認

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の管理ユーザ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（管理）設定を更新します。このコマンドは、管理ユーザを作成、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**

## ローカルユーザデータベースの更新とダウングレード

20.0 よりも前のリリースでは、ローカルユーザのパスワードは MD5 メッセージ ダイジェスト アルゴリズムでハッシュされ、ローカルユーザデータベースに保存されていました。リリース 20. 0 では、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、ローカルユーザデータベース内の既存のユーザパスワードは 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。

データベースのアップグレードプロセスでは、MD5 ハッシュパスワードが PBKDF2 形式に自動的に変換されることはありません。StarOS は、古い暗号化アルゴリズムを使用してユーザを

認証し続けます。これは、「弱いハッシュ」フラグの古い暗号化アルゴリズムを使用してユーザにフラグを付けます。このフラグは、 EXEC モードの **show local-user [verbose]** CLI コマンドの出力に表示されます。ユーザがクレデンシャルを使用して再ログインすると、StarOS は MD5 アルゴリズムを使用して入力したパスワードが確認され、PBKDF2 アルゴリズムを使用して新しいハッシュが作成されて、その結果がデータベースに保存されます。その後、StarOS はそのユーザの「弱いハッシュ」フラグをクリアします。

**重要** ハッシュ関数は一方向であるため、PBKDF2 でハッシュされたパスワードを MD5 形式に変換することはできません。ローカルユーザデータベースは、20.0 よりも前の StarOS リリースに戻す前にダウングレードする必要があります。

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、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用の TACACS+ の設定

ここでは、VPC-DI の TACACS+（Terminal Access Controller Access Control System+）AAA（認証、許可、およびアカウンティング）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+サーバにリモートでアクセスすることで、VPC-DI システムはシステム管理ユーザに 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（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、サーバへの接続が終了します。

TACACS+ は、VPC-DI 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** ユーザは、StarOS と TACACS+ サーバ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** *15.0 MR4* よりも後のリリースの場合、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に対しては生成されません。

# ユーザアカウントの要件

TACACS+ AAA サービスを設定する前に、次の TACACS+ サーバと StarOS ユーザ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+ ユーザアカウントの要件

TACACS+ サーバは、次の TACACS+ ユーザ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。

- 既知の管理ユーザのリスト。
- 各ユーザのプレーンテキストまたは暗号化されたパスワード。
- 各ユーザが属するグループの名前。
- ユーザグループのリスト。
- 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 リリース 20.0 以降の信頼できる StarOS のビルドでは、FTP はサポートされていません。

## StarOS ユーザアカウントの要件

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

- ユーザ名
- パスワード
- 管理ロールと権限

重要 システムでユーザ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*：1 ～ 3 の整数（*18.2* よりも前のリリース）または 1 ～ 4 の整数（*18.2+* のリリース）である必要があります。これは、この TACACS+サーバが TACACS+ 認証を使用する順序を指定します。優先順位は 1 が最も高く、3 または 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+ サーバに対応します。
- **ip-address**：有効な TACACS+ サーバの 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、およびアカウンティング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 （130 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、ユーザがログインを試行する VPN コンテキスト名は、ユーザ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-unkown-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、およびアカウンティング）サービス、およびローカルユーザに対して拡張されます。たとえば、ローカルユーザは、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。ユーザが TACACS+ ユーザでない場合、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** 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ログインの制限

StarOS ユーザを作成するときのセキュリティ管理者は、そのユーザ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、ローカルユーザデータベースの更新とダウングレード（122 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 ベースのユーザのコンソールアクセスの制限

AAA ベースのユーザは、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。

リリース 15.0 以降では、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。

リリース 19.2 以降では、ユーザはクイック 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 リリース 19.2 以降では、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。

リリース 15.0 以降では、より安全なアルゴリズムを使用してシャーシキーからシャーシ ID が生成されます。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。

リリース 14 とリリース 15 のシャーシ ID の形式は異なります。リリース 15 では、リリース 14 のシャーシ ID が認識され、有効と見なされます。14.x から 15.0 へのアップグレードでは、シャーシ ID または設定ファイルを変更する必要はありません。

ただし、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リリース15でシャーシキーをリセットすると、新しいシャーシ ID がリリース 15 形式（16 文字ではなく 44）で生成されます。リリース 14 のビルドでは、44 文字のシャーシ ID は認識されません。その後でシャーシをリリース 14 にダウングレードすると、新たに 16 文字のシャーシ ID が生成されます。古いキー形式に対応するには、ダウングレード前に設定ファイルを v12.2 以前の形式で保存する必要があります。ダウングレードされたシャーシにv15設定ファイルをロードしようとすると、StarOS は設定ファイルに保存されているパスワード/秘密を復号できなくなります。

リリース 19.2 以降では、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

クイック セットアップ ウィザードでは、ユーザにシャーシキーの値を入力するように求めるプロンプトが表示されます。シャーシキーの値が入力されていない場合は、シャーシの MAC アドレス（20.0 より前のリリース）を使用してデフォルトのシャーシが生成されます。

リリース 20.0 以降では、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。

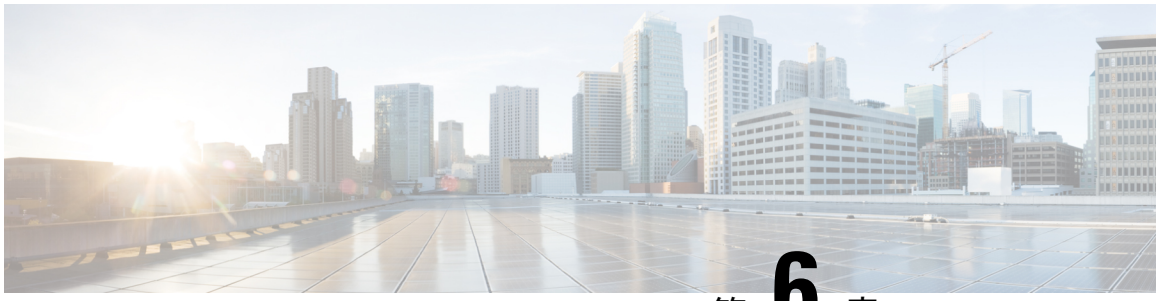
グローバル コンフィギュレーション モードの **fabric fsc-auto-recover** コマンドは、過剰な数の破棄されたファブリック出力パケットを検出したときに、自動 FSC リセットを有効または無効にします。

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 6 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

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 モードのロッキング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（140 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ったり、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（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）その後に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す。

# 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（141 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られます。

ユーザ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。

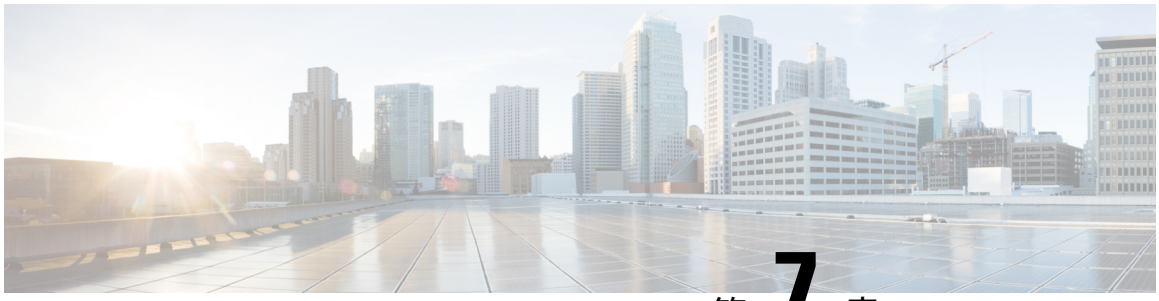
# show administrators コマンド

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 7 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

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詳しくは、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**

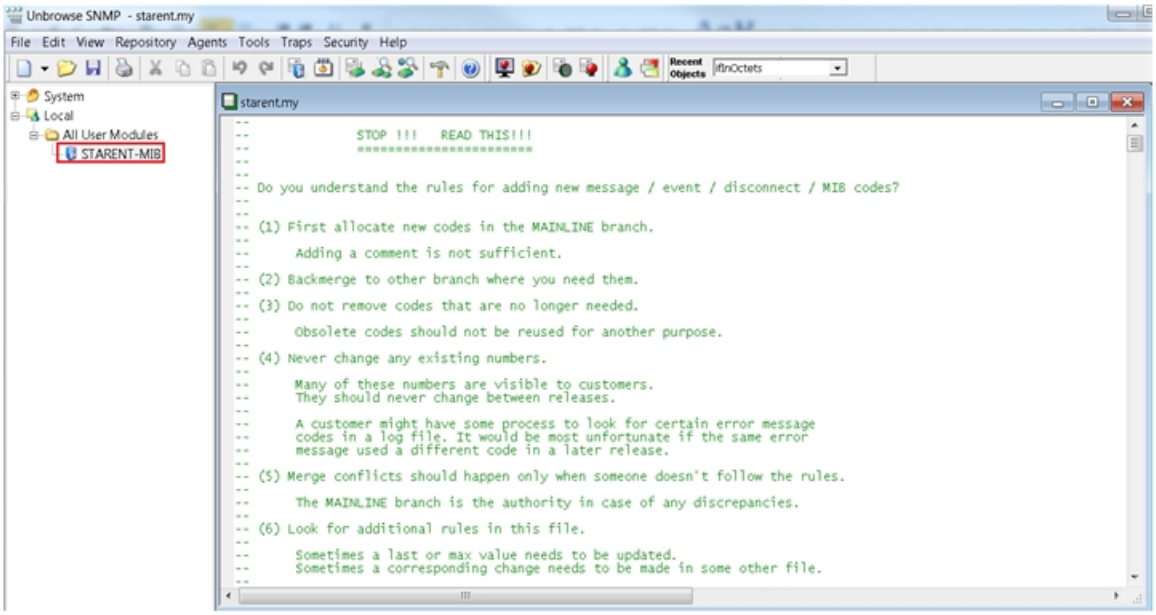
**ステップ 3** コンパニオンファイルを開き、解凍して、同じフォルダに展開します。

**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。

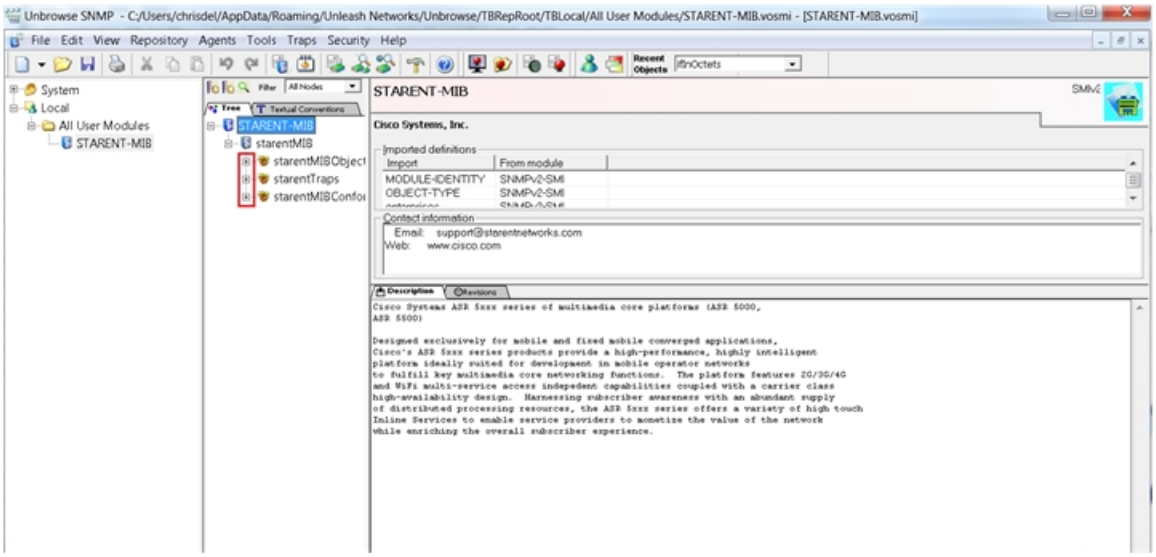
**ステップ 6** MIB ブラウザで、.tar ファイル内の **starent.my** ファイルを検索して開きます。SNMP MIB ブラウザを使用すると、MIB **.my** ファイルを表示する前にコンパイルできます。

**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。



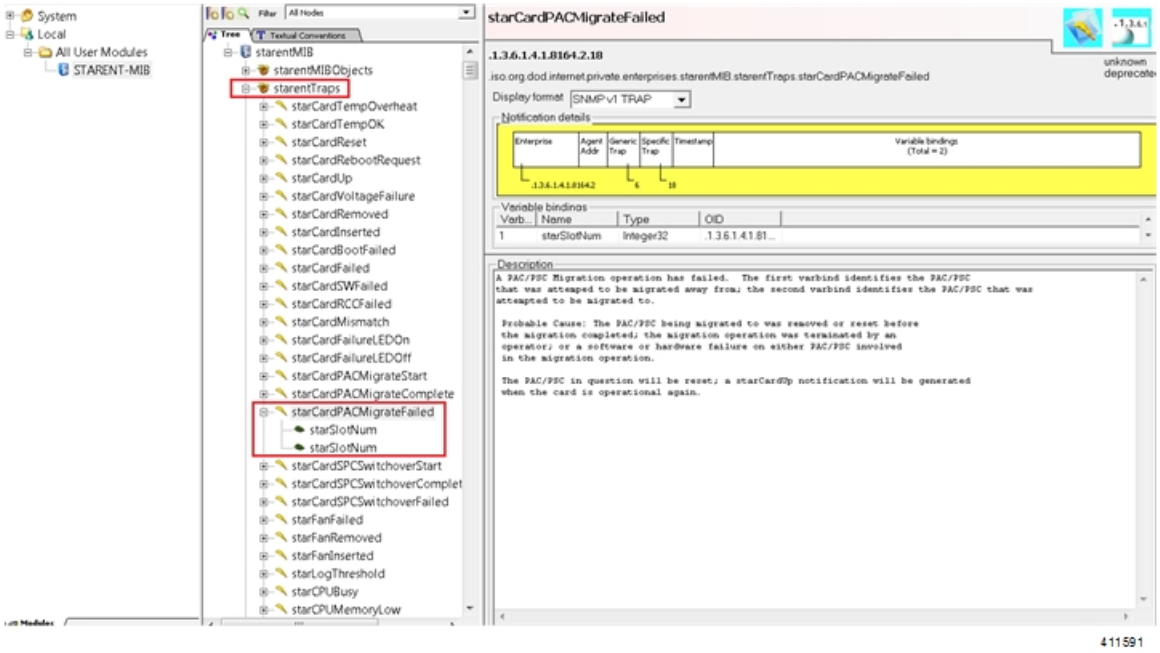

335998

STARENT-MIB.vosmi ファイルが開きます。




335999

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。



411591

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 サーバまたはアラームサーバ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 サーバまたはアラームサーバ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 とアラームサーバパラメータの設定 （146 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 （147 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 とアラームサーバパラメータの設定

SNMP およびアラームサーバ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**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。

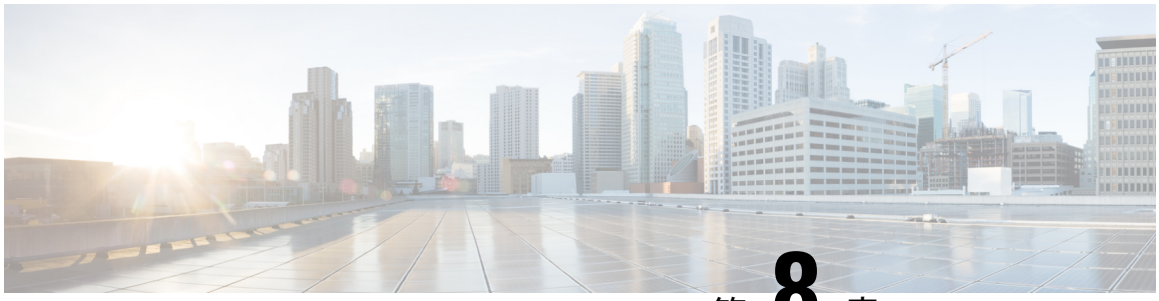
SNMP トラップの生成を設定するには、次の手順を実行します。

**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 8 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

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                                 : egtp1
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

**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

アクティブ CF でコンフィギュレーションまたは StarOS バージョンのブート順序が変更されるたびに、ファイルシステムをスタンバイ CF と同期する必要があります。これにより、管理カード間で変更が同一に維持されることが保証されます。

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 **-redundant** キーワードを指定すると、スタンバイ CF 仮想マシンにコンフィギュレーション ファイルが保存されます。このコマンドは、ローカルファイルシステムを同期しません。アクティブな CF VM のローカルデバイスとの間で他のファイルやディレクトリを追加、変更、または削除した場合は、両方の CF VM でローカルファイルシステムを同期する必要があります。ファイルシステムの同期 （151 ページ） を参照してください。

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。

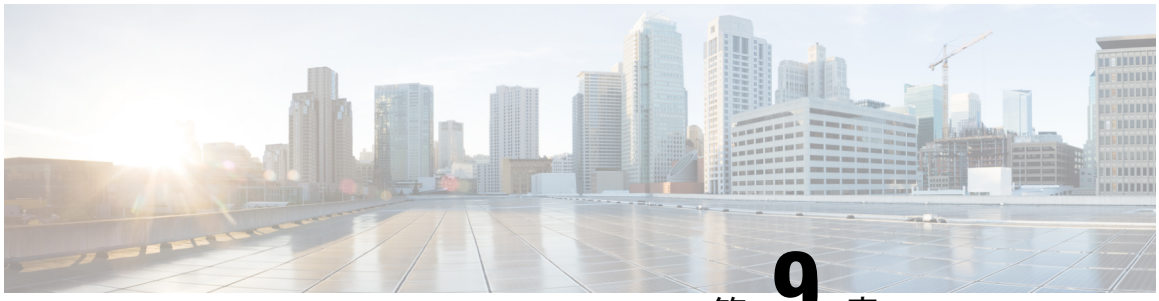
（注） このコマンドでは usb1 キーワードオプションおよび usb2 キーワードオプションを使用できますが、このオプションは、デバイスがハイパーバイザを介してサーバに設定されている場合にのみ使用できます。これには、仮想コントローラの作成と使用可能なデバイスの指定が含まれます。

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。

*system.cfg* というコンフィギュレーション ファイルを、あらかじめ *cfgfiles* という名前で作成しておいたディレクトリに保存するには、次のコマンドを入力します。

```
save configuration /flash/cfgfiles/system.cfg
```



第 9 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

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 （155 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 （155 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 （156 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：

- **ポートイーサネット** *slot#* の場合、スロットは仮想シャーシ内の CF または SV VM に対応します。ハイパーバイザは VPC-DI インスタンスの初期設定時に、各 VM に一意のスロット番号を割り当てます。スロット番号 1 および 2 は CF に割り当てられ、スロット番号 3 ～ 32 は SF に割り当てられます。

- **ポートイーサネット** *port#* の場合、CF はポート 1 のみをサポートします。各 SF は 10 ～ 21 の番号が付いた対応する仮想イーサネットポートを使用して、12 個の vNICs 番号 1 ～ 12 をサポートします。SF ポート番号 10 を設定する必要があります。

（注） StarOS は最大 12 個のサービスポートをサポートしますが、実際のポート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。

- 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 0.0.0.0 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#
```

*slot* はシャーシのスロット番号です。スロットは、仮想シャーシ内の CF または SF VM に対応します。ハイパーバイザは VPC-DI インスタンスの初期設定時に、各 VM に一意のスロット番号を割り当てます。スロット番号 1 と 2 は CF VM に割り当てられ、スロット番号 3 ～ 32 は SF VM に割り当てられます。CF はポート 1 のみをサポートします。各 SF は、1 ～ 4 番の 4 つの vNIC をサポートし、対応する仮想イーサネットポート 10 ～ 14 番を使用します。SF ポート番号 10 を設定する必要があります。

**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、スロット 3 の SF のポート 10 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 *sf3* 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 3/10
      description 10_SF3
      no shutdown
      bind interface sf3 source
      end
```

*slot*は、シャーシのスロット番号（常にスロット 1）です。

**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、192.168.250.1 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        192.168.250.1    Static      0    0     vNIC1
0.0.0.0/0        192.168.250.1    Static      0    0     sf3 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。

# VLAN

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。

特定のパケット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パケットを伝送します。

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。

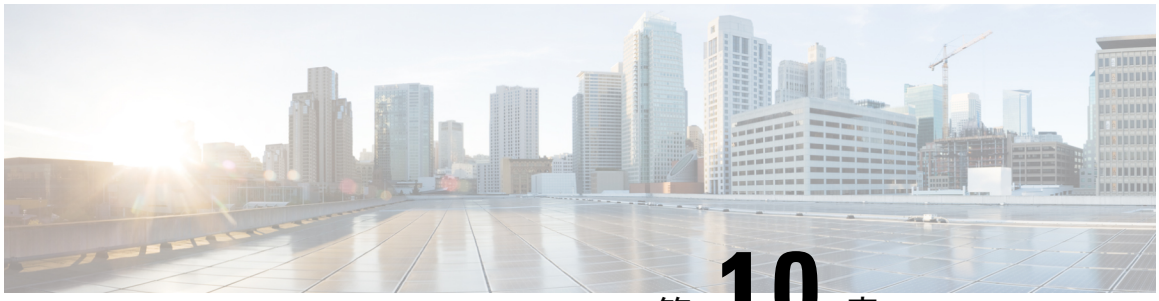
また、別の VLAN で他の OA&M サービスを設定することもできます。

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 10 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

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。シャーシキーは暗号キーとして使用されます。

シャーシキーを設定すると、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シ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、シャーシ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** リリース 21.0 以降では、デフォルトはアルゴリズム B です。

リリース 15.0 以降では、別のタイプの暗号化アルゴリズムを指定できます。グローバル コンフィギュレーション モードの **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。

リリース 19.2 以降では、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。

また、リリース 19.2 以降では、暗号キーはシャーシ ID および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シ 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シ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シ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シキーとランダムなソルトを使用します。

ユーザ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シキーを変更するための推奨プロセスは、次のとおりです。

- 前のシャーシキーを使用して、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シキーを変更します。
- この新しいシャーシキー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「シャーシキーの設定」を参照してください。

# ICSR 設定のサポート

シャーシ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-DIシャーシ/インスタンスを採用する冗長構成です。

ICSR ペアは同じシャーシキーを共有します。2 つのシャーシ/インスタンスに互換性のないシャーシキーがあることをISCR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。一致するシャーシキーがないと、アクティブピアがアウトオブサービスなった場合に、スタンバイ ICSR ピアはサービスをリカバリできます。スタンバイピアは、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 ピアは、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 サーバアドレスのリストを送信します。そのサブスクライ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、ユーザは、接続先の LI サーバ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 サーバアドレスに関連する情報が受信されない場合、LI サーバ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。

# ユーザの追加、変更、削除

システムに対するユーザ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を追加または削除したり、権限を上げたり、他のユーザ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、権限の低いユーザが権限を上げたり、権限の高いユーザ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、これらのユーザがシステムを介して設定されている場合にのみ、ユーザおよびユーザ属性（権限レベルなど）の変更を検出できます。

## ユーザの追加または削除の通知

承認レベルの低いユーザは、承認レベルの高いユーザを作成できません。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を作成できる場合は、他の高レベルの認可されたユーザを削除して、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 シェルへのユーザアクセス

**starOsShellAccessed** トラップは、ユーザ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、Coleto 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、 Coleto 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 (qvpc-si-medium)
  Crypto Accelerator      : Coleto 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, 1af4:0001 (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 | • VPC-DI<br>• 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 | • *VPC-DI* システム管理ガイド<br>• *VPC-SI System* アドミニストレーション 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 |
|---|---|
| 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。

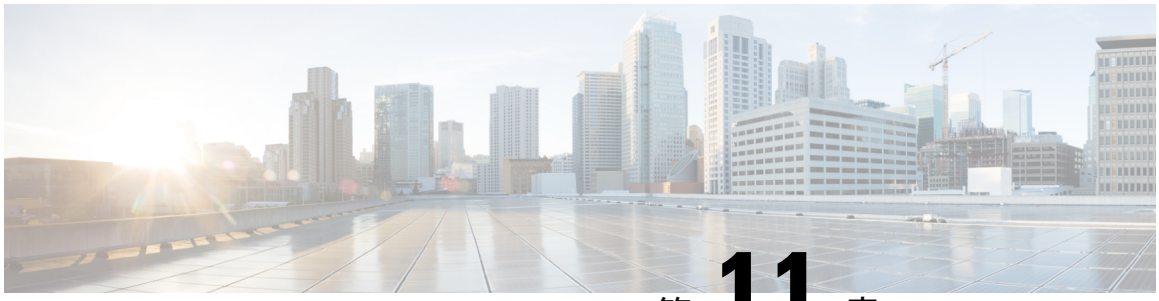
HWRNGサポートが使用可能な場合は、ホスト上の `libvirt xml` ファイルに次の設定を追加します。これにより、クライアント（StarOS）に `virtio_rng` サポートが追加されます。

```
<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 11 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

**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<br>VPC-DI<br>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 | • *ASR 5500 System Administration Guide*<br>• *VPC-DI* システム管理ガイド<br>• *VPC-SI System* アドミニストレーション ガイド |

**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 （176 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

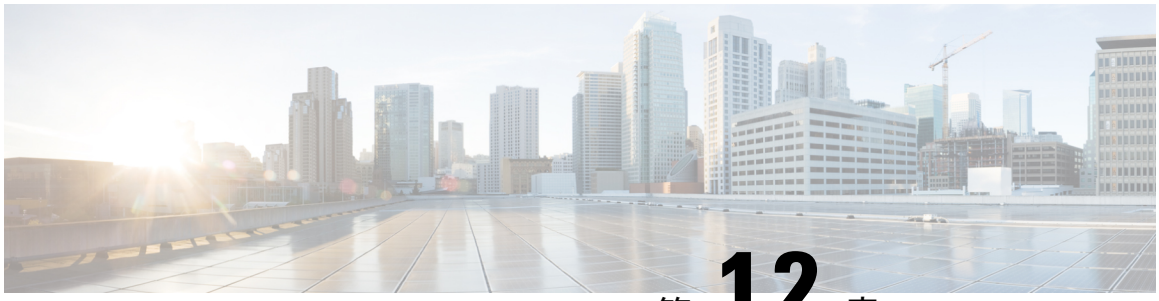
コンフィギュレーションファイルで署名の検証を有効化（または無効化）するには、次のコマンドを使用します。

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 12 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

VPC VM 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す。

- **/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 ウィザードの使用方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。

（注） 次に説明するコマンドを実行するには、アクティブ CF VM にログインする必要があります。

# 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_confg.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とインスペクタレベルのユーザは **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
- 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
```

# 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。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（178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、システムの起動方式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とインスペクタレベルのユーザが実行できます。

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** デバイスに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スキームがリリース 16.1 で変更されました。16.1 よりも前の形式は「production.*image*.bin」です。16.1 以降の形式は、「asr5500-*image_number*.bin」となっています。この変更は、以下に示す例に反映されています。

**例 1**：*16.1* よりも前の *StarOS* リリース：

```
boot system priority 18 \
    image /flash/15-0-builds/production.45666.bin \
    config /flash/general_config.cfg

boot system priority 19 \
    image /flash/15-0-builds/production.45717.bin \
    config /flash/general_config_3819.cfg

boot system priority 20 \
    image /flash/15-0-builds/production.45069.bin \
    config /flash/general_config_3665.cfg
```

**例 2**：*StarOS* リリース *16.1*以降：

```
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 \config
/flash/general_config.cfg
boot system priority 19 image tftp://192.168.1.161/tftpboot/image_version.bin \config
/flash/general_config.cfg
boot system priority 20 image /flash/image_version.bin \config /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（186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。

**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. OS リリースバージョンとビルド番号の識別 （187 ページ）

2. 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 （188 ページ）
3. /flash デバイスの空き領域の確認 （188 ページ）
4. StarOS イメージを /flash に転送 （189 ページ）
5. 実行コンフィギュレーションの保存 （189 ページ）
6. ファイルシステムの同期 （191 ページ）
7. システムの再起動 （192 ページ）

# 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。

*20.0* 以降の *StarOS* リリースでは、リリース前に REL キーを使用して starfile イメージに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。

*20.0* 以降の *StarOS* リリースでは、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

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. 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**

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。リリース 20.0 以降の信頼できる StarOS のビルドでは、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。

# 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。

# 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. で取得した 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
```

# 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. 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. で取得した 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. 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**

## 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|0000000000000|
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。

**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と、**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|0000000000000|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 ] ]**

# 新しいライセンスキーのアクティブ化

新しいライセンスキーをアクティブにするには、Exec モードで **reload** コマンドを実行してアクティブな CF を再起動する必要があります。

（注） **reload** コマンドがアクティブ CF で実行されると、すべての SF が再起動されます。

# 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 （131 ページ）およびコンソール/VTY 回線でのローカルユーザログインの制限 （131 ページ）を参照してください。

## ローカルユーザパスワードのプロパティの設定

ローカルユーザ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。

## ローカルユーザのアカウント管理プロパティの設定

ローカルユーザ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の一時停止が含まれています。

### ローカルユーザアカウントのロックアウト

ローカルユーザアカウントは、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。
- **パスワードのエージング**：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）、ユーザはアクセスできません。

**重要** ローカルユーザの管理ユーザアカウントは、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。

## ローカルユーザアカウントの一時停止

ローカルユーザアカウントは、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
```

# ローカルユーザパスワードの変更

ローカルユーザの管理ユーザは、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のパスワードをリセットすると、ユーザ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。

# VPC-DI インスタンスでの VM のリセット、停止、開始、または削除

ハイパーバイザを使用して、VPC-DI インスタンス内の 1 つ以上の VM をリセット、停止、開始、または削除するスクリプトを作成できます。

## KVM

指定された VM をリセット、停止、開始、または削除するための一連のコマンドを実行する KVM スクリプトを作成できます。

実際のスクリプトには、サイトに展開されている VPC-DI インスタンスのインストール要件に適したコマンドシーケンスが含まれています。

### スクリプトコマンド

使用可能なコマンドは次のとおりです。

```
 reset
for CARD in {01,02}-cf {03,04,05,06...32}-sf do
  VMNAME={NAME}{INST}-{CARD}
  virsh reset {VMNAME}
done

 stop
for CARD in {01,02}-cf {03,04,05,06...32}-sf do
  VMNAME={NAME}{INST}-{CARD}
  virsh destroy {VMNAME}
done

 start
for CARD in {01,02}-cf {03,04,05,06...32}-sf do
  VMNAME={NAME}{INST}-{CARD}
  virsh start {VMNAME}
done

 delete
for CARD in {01,02}-cf {03,04,05,06...32}-sf do
  VMNAME={NAME}{INST}-{CARD}
  virsh undefine {VMNAME}
done
```

### サンプルスクリプト

サンプルスクリプトを次に示します。

```
cat ./vm-start

 start local
for CARD in {01,02}-cf {03,04,05,06,17,18,19,20}-sf do
  VMNAME=BLADE5-DI-{CARD}
  virsh start {VMNAME}
done
```

```
 start remote
ssh testbed3 '/home/luser/vm-start-BLADE5-DI'

roottestc1b5:/home/luser cat ./vm-stop

 reset local
for CARD in {01,02}-cf {03,04,05,06,17,18,19,20}-sf do
  VMNAME=BLADE5-DI-{CARD}
  virsh destroy {VMNAME}
done

 reset remote
ssh testbed3 '/home/luser/vm-stop-BLADE5-DI'
```

# VMware ESXi

## vSphere GUI

VPC-DI インスタンス内の VM をリセットするには、vSphere GUI を使用します。

**ステップ 1** vSphere から、ターゲット VPC-DI VM のホストを選択します。

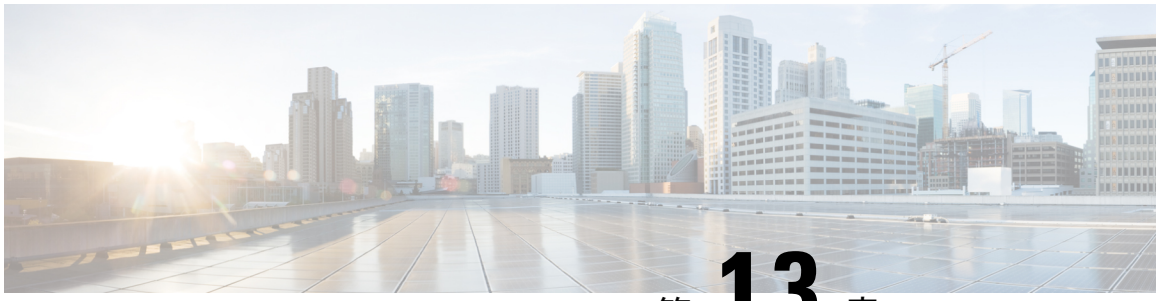
**ステップ 2** [Commands] リストから [Power Off] を選択して、VM をシャットダウンします。

**ステップ 3** [Commands] リストから [Power On] を選択して VM を起動します。

**ステップ 4** この VPC-DI インスタンスの VM をプロビジョニングするために複数の vSphere ホストが使用されている場合は、ホストごとにステップ 2 と 3 を繰り返します。

## PowerShell スクリプト

また、XREF で説明されているコマンドを使用してターゲット VM を再起動する PowerShell スクリプトを作成することもできます。



第 13 章

# スマートライセンス



## 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<br>VPC-SI<br>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 | • *ASR 5500 System Administration Guide*<br>• *Command Line Interface Reference*<br>• *VPC-DI* システム管理ガイド<br>• *VPC-SI System* アドミニストレーション ガイド |

**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 ライセンシング

## 概要

StarOS 21.3 以降のリリースでは、スマートライセンシングがサポートされています。スマートライセンシングは、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シ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。スマートライセンシ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 ライセンシ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。

## 従来のライセンシングとスマートライセンシングの比較

シスコでは、従来のライセンシングとスマート ソフトウェア ライセンシングの 2 種類のライセンスモデルを採用しています。**従来のライセンシング**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 ライセンシング**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シングです。StarOS に組み込まれたスマート ソフトウェア ライセンシング機能により、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、ライセンス供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、シスコライセンスの購入の追跡を行えます。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** クレデンシャ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、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帰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<br>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<br>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<br>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<br>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<br>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<br>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<br>GW バンドル、10K セッションの提供 | regid.2017-02.com.cisco.<br>ASR5K-00-SW10LIC, 1.0_ca9d09ee-<br>5e72-4695-97ef-d5117037cfb4 |
| 4 | ASR5K-00-ME10LIC<br>MME バンドル、10K セッション | regid.2017-02.com.cisco.<br>ASR5K-00-ME10LIC, 1.0_<br>37a452c9-ff23-41fc-b96c-c8b2eb09254b |
| 5 | ASR5K-00-EG10S-K9<br>EPDG、10k セッション | regid.2017-02.com.cisco.<br>ASR5K-00-EG10S-K9,<br>1.0_5c625d0f-8b2c-4c44-970b- 9668a072890f |
| 6 | ASR5K-00-SM10S-K9<br>SaMOG GW 10k セッション | regid.2017-02.com.cisco.<br>ASR5K-00-SM10S-K9,<br>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<br>GRE インターフェイス トンネリング | regid.2017-11.com.cisco.<br>ASR5K-00-CS01GRET,<br>1.0_284ba242-a2ef-4ad0-b445-b62ed3887d3b |
| 1 | ASR5K-00-CS01MPLS<br>StarOS 上の MPLS | regid.2017-11.com.cisco.<br>ASR5K-00-CS01MPLS,<br>1.0_1b1221cd-39de-4113-8709-dad022c0a063 |
| 2 | ASR5K-00-GN01REC<br>セッションリカバリ | regid.2017-11.com.cisco.<br>ASR5K-00-GN01REC,<br>1.0_88542f00-077b-4ed0-9698-a3cec699a047 |
| 3 | ASR5K-00-GN01ICSR<br>シャーシ間セッションリカバリ | regid.2017-11.com.cisco.<br>ASR5K-00-GN01ICSR,<br>1.0_d0bca3e3-1218-4e91-98f6-207a20a5401a |
| 4 | ASR5K-00-EP01VLE<br>EPC Gw VoLTE の機能拡張 | regid.2017-11.com.cisco.<br>ASR5K-00-EP01VLE,<br>1.0_a6c3d615-0b37-4f4a-8c34-3090ed5e265c |
| 5 | ASR5K-00-EP01ICSE<br>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.<br>ASR5K-00-EP01ICSE,<br>1.0_bf87e837-003e-4832-b29f-ce0416699b01 |
| 6 | ASR5K-00-PWXICSRP<br>ICSR/SR のパフォーマンスの向上 | regid.2017-11.com.cisco.<br>ASR5K-00-PWXICSRP,<br>1.0_079a8bd2-a97d-4d12-bc5a-b0f70eec6c26 |

| TagId | 機能の説明 | 権限付与タグ |
|---|---|---|
| 7 | ASR5K-00-CS01I-K9<br>StarOS 上の IPSec | regid.2017-11.com.cisco.<br>ASR5K-00-CS01I-K9,<br>1.0_a6e5539c-ab69-4303-b2a8-e71c03ac2a08 |
| 8 | ASR5K-00-EG01RS<br>ePDG Reselection | regid.2017-11.com.cisco. ASR5K-00-EG01RS,<br>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 / starSmartLicenseServiceOOCClear
- 1350：starSmartLicenseEvalMode/starSmartLicenseEvalModeClear
- 1351：starSmartLicenseEvalModeExpire/starSmartLicenseEvalModeExpireClear
- 1352：starSmartLicenseCSSMConntectionFail/starSmartLicenseCSSMConnectionFailClear

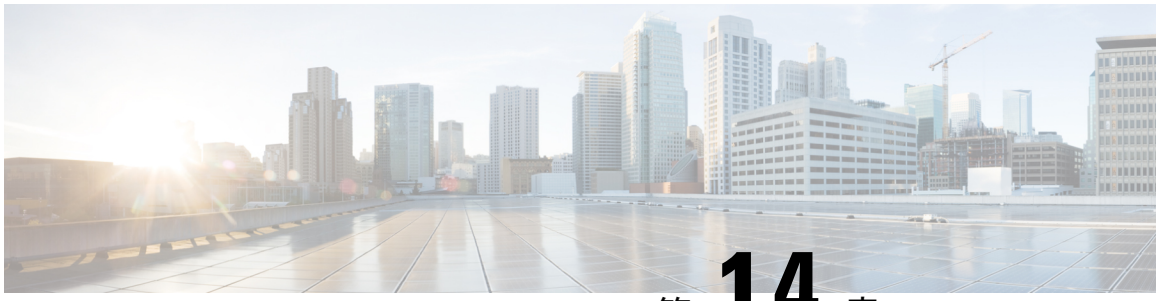
（注） これらの SNMP MIB アラームの詳細については、『*SNMP MIB Reference*』を参照してください。

# 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 14 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）サーバを認識しません。ハイパーバイザと COTS サーバのステータスをモニタするには、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

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。 |
|---|---|
| **管理情報を表示** | |
| 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のリストを表示 | **show administrators** |
| 管理ユーザが作業しているコンテキスト、管理ユーザが 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 | **show administrators session id** |
| システムに設定されているローカルユーザ管理アカウントに関連する情報を表示 | **show local-user verbose** |
| ローカルユーザ管理アカウントの統計情報を表示 | **show local-user statistics verbose** |
| CLI セッションに関連する情報を表示 | **show cli** |
| **システム稼働時間を決定** | |
| システム稼働時間を表示（前回の再起動からの経過時間） | **show system uptime** |
| **NTP サーバのステータスを表示** | |
| NTP サーバのステータスを表示 | **show ntp status** |
| **システムリソースを表示** | |
| 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） | **show resources [ cpu ]** |
| **システムアラームを表示** | |
| 現在未処理のすべてのアラームに関する情報を表示 | **show alarm outstanding all verbose** |

| 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 | **show alarm statistics** |
| **輻輳制御の統計情報を表示** | |
| 輻輳制御の統計情報を表示 | **show congestion-control statistics** |
| **リモート管理の統計情報を表示** | |
| SNMP 通知の統計情報を表示 | **show snmp notifies** |
| SNMP アクセスの統計情報を表示 | **show snmp accesses** |
| SNMP トラップの履歴を表示 | **show snmp trap history** |
| SNMP トラップの統計情報を表示 | **show snmp trap statistics** |
| **ポートカウンタを表示** | |
| 特定のポートのデータリンクカウンタを表示 | **show port datalink counters** *slot#/port#* |
| 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 | **show port npu counters** *slot#/port#* |
| **システム情報とネットワークインターフェイスを表示** | |
| 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 | **show hardware** |
| **カード情報と統計情報を表示** | |
| 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 | **show card diag** *slot/port* |
| 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 | **show card info** *slot/port* |
| すべてのカードまたは VM の動作ステータスを表示 | **show card table** |
| 起動設定（param.cfg）ファイルのコンテツを表示<br>[VPC-DI] | **show cloud configuration** |
| 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<br>[VPC-DI] | **show cloud hardware** |
| 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対象統計情報を表示<br>[VPC-DI] | **show cloud monitor di-network** |

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。

# DI ネットワークのモニタリング

DI ネットワークは、VM をインターコネクトするプライベート L2 ネットワークです。DI ネットワークは、受信した VM から別の VM 上でサービスを提供するセッションマネージャにユーザトラフィックを転送します。また、CLI コマンド、正常性チェック、ステータス変更などの SF 通信にも CF を転送します。リンクが侵害されると、予期しない事態（CLI コマンドへの応答が遅いなど）が発生し、サービスが中断される可能性があります。

DI ネットワークの正常性を確認するために使用可能なモニタリング機能は、次のとおりです。

### SF 間の DI ネットワークテスト

各 SF は、非ブロッキング UDP テストパケットを他のアクティブ SF とスタンバイ SF のそれぞれに定期的に送信し、応答を追跡し続けて遅延とパケット損失を計算します。テストパケットは 1 秒に 1 回送信されます。ジャンボと非ジャンボの両方のテストパケットは、交互に送信されます。非ジャンボ UDP テストパケットのペイロードサイズは 200 バイトで、ジャンボテストパケットのペイロードサイズは 4000 バイトです。統計情報が記録されます。

- ドロップされたパケット数：別の SF からのテストパケットを受信すると、受信 SF は応答を返します。SF が 1 秒以内にテストパケット応答を受信しなかった場合、パケットはドロップされたものとしてマークされます。
- ドロップされたジャンボパケット：ドロップされたパケット数と同じ計算ですが、ジャンボテストパケットのみをカウントします。
- 長時間の遅延が発生しているパケットの数：SF が 200 ミリ秒後にテストパケット応答を受信した場合、パケットに長い遅延があるとマークします。

（注） カウンタは、SF の再起動後にクリアされます。

レポート間隔は 15 秒で開始され、3600 秒のまでの範囲で指定できます。間隔の間にエラーが検出されなかった場合は、警告ログは生成されず、間隔が 3600 秒になるまでレポート間隔が倍増します。間隔の間にエラーが検出されると、警告ログが生成され、ドロップされたパケットがなくなるまで、レポート間隔が半減します。

パケット損失または長時間の遅延が発生した場合は、警告イベントが生成されます。次に、警告の例を示します。

```
2016-Jan-10+22:00:01.477 [hat 3081 warning] [5/0/5146 <hatcpu:50> hatcpu.c:1307] [software

internal system syslog] Over the past 15 seconds, tests from card 5 to 4 had 1 total
drops,
0 jumbo drops, 0 long latency.
```

SF 間通信に関して収集された統計情報を表示するには、**show heartbeat stats card***cardnumber* **cpu** *cpunumber* コマンドを使用します。

DI ネットワークのモニタリングはデフォルトでは有効になっています。特定の SF 上の SF パケットテストを停止して開始するか、または特定の SF 上でテストパケットカウンタをクリアするには、**debug heartbeat test** コマンドを使用します。

また、**show cloud monitor di-network** コマンドを使用して、DI ネットワークモニタリングの統計情報を表示することもできます。次に、カード番号 3 での **show cloud monitor di-network summary** コマンドの出力例を示します。

```
Card 3 Test Results:

ToCard   Health   5MinLoss   60MinLoss
   1      Good       0.0%       0.0%
   2      Good       0.0%       0.0%
   4       Bad       6.32%      5.36%
   5      Good       0.0%       0.0%
   6      Good       0.0%       0.0%
```

この表示には、過去 5 分間および過去 60 分間のテストパケット損失率が示されます。この率が 1% を超えると、正常性ステータスが「Bad」としてマークされます。

### SF からスタンバイ CF への DI ネットワークテスト

SF の起動中に、各 SF は非ジャンボ ping パケットおよびジャンボ ping パケットの両方をスタンバイ CF に送信して、スタンバイ CF が到達可能であることを確認します。

SF の通常動作中に、SF は非ブロッキング UDP テストパケットをスタンバイ CF に定期的に送信し、応答を追跡し続けて遅延とパケット損失を計算します。11965 この機能は、*SF* 間の *DI* ネットワークテストで説明されているものと同じです。

### SF セカンダリ IP アドレスの DI ネットワークテスト

SF の起動中に、各 SF は、SF プライマリ IP アドレスを使用して、非ジャンボ ping パケットおよびジャンボ ping パケットの両方をアクティブ CF に送信します。さらに、各 SF は、それぞれのセカンダリ IP アドレスを使用して、非ジャンボ ping パケットをアクティブ CF に送信します。これらの ping のいずれかが失敗すると、SF はアクティブ CF に通知し、SF が再起動します。

### スタンバイ CF からアクティブ CF への DI ネットワークテスト

スタンバイ CF の起動時に、スタンバイ CF は、非ジャンボ ping パケットおよびジャンボ ping パケットの両方をアクティブ CF に送信します。

### DI: ネットワークのバルク統計情報

**mon-di-net** スキーマは、VPC-DI プラットフォーム上の DI ネットワークの正常性をモニタリングするために、次のバルク統計情報を提供します。この情報は、Exec モードの **show cloud monitor di-network summary** コマンドの出力で提供されるものと類似しています。

- src-card：モニタリングが実行された送信元カードのスロット番号。
- dest-card ；トラフィックがルーティングされた接続先カードのスロット番号。
- total-pkts-5mins ：過去 5 分間に送信されたパケットの合計数。
- total-drops-5mins：過去 5 分間にドロップされたパケットの合計数。
- total-pkts-60mins：過去 60 分間に送信されたパケットの合計数。
- total-drops-60mins：過去 60 分間にドロップされたパケットの合計数。
- total-pkts：送信されたすべてのパケットの合計数。
- total-pkts-jumbo ：送信されたジャンボパケットの合計数。
- total-drops ：ドロップされたジャンボテストパケットと非ジャンボテストパケットの合計数。
- total-drops-jumbo ： ドロップされたジャンボテストパケットの数。
- latency-warnings：遅延がしきい値を超えた回数の合計。
- long-rtt ：ミリ秒単位の最長ラウンドトリップ時間（RTT）。
- average-rtt：見ろ秒単位の平均ラウンドトリップ時間（RTT）。

統計情報モードの **mon-di-net** コマンドは、Mon-DI-Net スキーマの統計情報の収集を設定します。

バルク統計情報収集の設定については、「バルク統計情報」の章を参照してください。

### DI ネットワークのハートビートのしきい値

この機能により、VPC-DI の展開でカード間のネットワークの正常性をモニタするために、ハートビート損失の割合で内部 DI ネットワークのしきい値を定義する機能が追加されます。

（いずれかのカード上の）ハートビート損失がしきい値の制限を超えた場合、この機能によってアラーム/SNMP トラップが生成され、その損失が示されます。

内部高可用性タスク（HAT）は、過去 5 分間のハートビート損失とカード間の 60 分間の割合を追跡し、しきい値を超えた場合、または前のアラームがクリアされた場合に SNMP アラームを生成できます。

システムには複数のカードがあり、どのカードも同じトラップ ID を生成できますが、カード情報は異なる場合があります。

この機能の範囲はシステム全体に及びます。これはサービスに固有ではなく、グローバル コンフィギュレーション モードで設定されます。

この機能を有効にする手順の詳細は、DI ネットワークハートビートのしきい値の設定 （223 ページ）を参照してください。

# モニタ VPC-DI ネットワーク

## 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 | 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 | • *Command Line Interface Reference*<br>• *VPC-DI* システム管理ガイド<br>• *Statistics and Counters Reference* |

### 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 よりも前 |

## 機能説明

DI ネットワークでは、DI ネットワークポートが飽和状態になっている場合、または基盤となるネットワーク インフラストラクチャが信頼できない場合にパケット損失が発生します。VPC-DI ネットワークのモニタ機能は、VPC-DI システムでのコントロールプレーンとデータプレーンのパケット損失の識別と定量化を可能にします。

VPC-DI は、CLI レポートとしきい値アラームで使用するコントロールプレーンとデータプレーンのモニタデータを収集して集約します。

また、この機能ではカードの障害を宣言するための VPC-DI の基準を設定することもできます。現在、アクティブな CF カードと SF カードの間に固定数のハイアベイラビリティタスク（HAT）コントロールプレーンのハートビートが連続してバウンスされると、カード障害が発生します。この機能を使用して、連続する失敗の数を設定できます。この機能により、セカン

ダリデータプレーンの設定パラメータが追加され、これを使用することで、DI ネットワークパケット損失とパケット処理障害のシナリオを効率的に区別できます。

## 機能の仕組み

コントロールプレーンとデータプレーンモニタは、2つの基本的なDIネットワークトラフィックタイプを固定または定期的に生成し、損失を追跡します。トラッキングデータは、DI ネットワーク通信の損失または中断を表示することを目的としています。

コントロールプレーンパケットは、通常、カード間の双方向 UDP/TCP ストリームをユニキャストします。基本的には、StarOS Proclet 間の要求と応答のペアです。

データプレーントラフィックは、カード間で転送されるユニキャストIPプロトコル254パケットで構成されます。このトラフィックは、StarOsが適切なアプリケーションインスタンス（入力）またはサービスポートインターフェイス（出力）に内部的に転送するサービスポートの入力または出力であり、確認応答されません（つまり、応答パケットはありません）。たとえば、SF5 で Session Manager インスタンスがサービスを提供する SF3 ポートに到着した入力パケットは、SF3 から SF5 への DI ネットワークを通過します。

すべての動作カード（つまり、動作状態がアクティブまたはスタンバイのCFおよびSF）は、モニタパケットを送受信します。モニタトラフィックは完全にメッシュ化されており、すべてのカードが他のすべてのカードにモニタパケットを送信し、他のすべてのカードからモニタパケットを受信します。

データプレーンパケットは、10/秒のレートで生成されます。コントロールプレーンモニタパケットは、5/秒のレートで生成されます。両方のパケットヘッダーに、デフォルトの優先順位が付けられています。

StarOSは、すべてのカード接続について、モニタの送信、受信、およびドロップデータを収集および集約します。**show cloud monitor controlplane** および **show cloud monitor dataplane** CLI コマンドは、現在の 15 秒、5 分、および 60 分のデータを表示します。5 分と 60 分の損失率は、bulkstat mon-di-net スキーマの変数として使用できます。5 分と 60 分の損失率は、しきい値アラームやトラップとしてもアクセス可能です。

ゼロ以外の破棄確率または低破棄確率は正常であることに注意してください。測定には、完全には同期していないカードペア間の相関関係が含まれているため、要求が生成されたのと隣接する間隔で応答を受信する可能性があります。これは、要求間隔のドロップとして反映されます。

一定の間隔で表示される場合、ドロップまたは損失率が高くなると、DI ネットワークの設定または動作上の問題、トラフィックの過負荷、または VM やホストの問題が発生することがあります。クラウドモニタは、DI ネットワークトラフィックの損失を表示し、特性を把握する機能を提供します。通常は、根本原因を特定するためにさらに調査する必要があります。

### 制限事項

モニタ VPC-DI ネットワーク機能には、次の制限事項があります。

- VPC-DI プラットフォームでのみサポートされています。
- ライセンス制御されていません。

# モニタ VPC-DI ネットワーク機能の設定

ここでは、機能を有効または無効にするために使用できるCLIコマンドについて説明します。

## カード障害検出の設定

セカンダリカードの障害検出基準を設定するには、次のコマンドを使用します。このコマンドはグローバル コンフィギュレーション モードで設定します。

```
configure
  high-availability fault-detection card dp-outage seconds
  end
```

注：

- **default**：デフォルトの dp-outage の値を復元します。デフォルト値は 2 秒です。
- dp-outageの延期は制限されていることに注意してください。連続するハートビートバウンスが設定されたhb-lossパラメータより5以上大きい場合、dp-outage設定に関係なく、カード障害が宣言されます。
- **dp-outage** パラメータは、VPC-DI プラットフォームでの管理者アクセスに制限されています。
- この CLI が設定されていない場合、デフォルトの dp-outage の値は 2 秒です。

## コントロールプレーンでのパケット損失しきい値の設定

次のコマンドを使用して、コントロールプレーン上の対応する時間間隔でのパケット損失率を測定します。しきい値アラームとSNMPトラップは、指定された期間に設定された損失率を超えるカード間接続に対して発生します。このコマンドはグローバル コンフィギュレーション モードで設定します。

```
configure
  [ default ] threshold cp-monitor-5min-loss pct [ clear pct ]
  end
```

```
[ default ] threshold poll cp-monitor-5min-loss interval duration
```

```
configure
  [default] threshold cp-monitor-60min-loss pct [ clear pct ]
  end
```

```
[default] threshold poll cp-monitor-60min-loss interval duration
```

注：

- **default**：コントロールプレーンの設定されたしきい値をクリアします。
- **clear** *pct*：設定されているパケット損失率をクリアします。
- **interval** *duration*：ポーリング間隔を構成する時間の長さ（秒単位）を指定します。*duration* は、60 ～ 60,000 の整数である必要があります。デフォルトは 300 秒です。
- このコマンドは、デフォルトでディセーブルになっています。

（注） この機能に関する補足情報については、『*Command Line Reference*』の「*Global Configuration Mode Commands*」の項を参照してください。

これらのしきい値を超過すると、次のアラームまたはトラップが生成されます。

- ThreshControlPlaneMonitor5MinsLoss / ThreshClearControlPlaneMonitor5MinsLoss
- ThreshControlPlaneMonitor60MinsLoss / ThreshControlPlaneMonitor60MinsLoss

これらのアラームまたはトラップの詳細については、『*SNMP MIB Reference*』を参照してください。

## データプレーンでのパケット損失しきい値の設定

次のコマンドを使用して、データプレーン上の対応する時間間隔でのパケット損失率を測定します。しきい値アラームと SNMP トラップは、指定された期間に設定された損失率を超えるカード間接続に対して発生します。このコマンドはグローバル コンフィギュレーション モードで設定します。

```
configure
   [ default ] threshold dp-monitor-5min-loss pct [ clear pct ]
   end
```

```
[ default ] threshold poll dp-monitor-5min-loss interval duration
```

```
configure
   [default] threshold dp-monitor-60min-loss pct [ clear pct ]
   end
```

```
[ default ] threshold poll dp-monitor-60min-loss interval duration
```

注：

- **default**：データプレーンの設定されたしきい値を無効にします。
- **clear** *pct* ：設定されたパケット損失をクリアします。
- **interval** *duration*：ポーリング間隔を構成する時間の長さ（秒単位）を指定します。*duration* は、60 ～ 60,000 の整数である必要があります。デフォルトは 300 秒です。
- このコマンドは、デフォルトでディセーブルになっています。

（注） この機能に関する補足情報については、『*Command Line Reference*』の「*Global Configuration Mode Commands*」の項を参照してください。

これらのしきい値を超過すると、次のアラームまたはトラップが生成されます。

- ThreshDataPlaneMonitor5MinsLoss / ThreshClearDataPlaneMonitor5MinsLoss
- ThreshDataPlaneMonitor60MinsLoss / ThreshDataPlaneMonitor60MinsLoss

これらのアラームまたはトラップの詳細については、『*SNMP MIB Reference*』を参照してください。

## 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で使用できるCLIコマンドについて説明します。

### 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コマンドまたはその出力について説明します。

#### show cloud monitor controlplane

この新しいshowコマンドは、最新のコントロールプレーンモニタ情報に対して次の出力を表示するために導入されました。

**show cloud monitor controlplane**

```
  Cards      15 Second Interval     5 Minute Interval    60 Minute Interval
 Src  Dst    Xmit   Recv  Miss%    Xmit   Recv  Miss%    Xmit   Recv  Miss%
 ---  ---  ------ ------ ------  ------ ------ ------  ------ ------ ------
  01   02      75     75   0.0%    1500   1500   0.0%   18000  18000   0.0%
  01   03      75     75   0.0%    1500   1500   0.0%   18000  17996   0.0%
  01   04      75     75   0.0%    1500   1500   0.0%   18000  18000   0.0%
  01   05      75     75   0.0%    1500   1500   0.0%   18000  18000   0.0%
  01   06      75     75   0.0%    1500   1500   0.0%   18000  18000   0.0%

  02   01      75     75   0.0%    1500   1500   0.0%   18000  18000   0.0%
  02   03      75     75   0.0%    1500   1500   0.0%   18000  17997   0.0%
  02   04      75     75   0.0%    1500   1500   0.0%   18000  18000   0.0%
  02   05      75     75   0.0%    1500   1500   0.0%   18000  18000   0.0%
  02   06      75     75   0.0%    1500   1500   0.0%   18000  18000   0.0%

  03   01      75     75   0.0%    1500   1500   0.0%      -incomplete-
  03   02      75     75   0.0%    1500   1500   0.0%      -incomplete-
  03   04      75     75   0.0%    1500   1500   0.0%      -incomplete-
  03   05      75     75   0.0%    1500   1500   0.0%      -incomplete-
  03   06      75     75   0.0%    1500   1500   0.0%      -incomplete-

  04   01      75     75   0.0%    1500   1500   0.0%   18000  18000   0.0%
  04   02      75     75   0.0%    1500   1500   0.0%   18000  18000   0.0%
  04   03      75     75   0.0%    1500   1500   0.0%   18000  17996   0.0%
  04   05      75     75   0.0%    1500   1500   0.0%   18000  18000   0.0%
  04   06      75     75   0.0%    1500   1500   0.0%   18000  18000   0.0%

  05   01      75     75   0.0%    1500   1500   0.0%   18000  18000   0.0%
  05   02      75     75   0.0%    1500   1500   0.0%   18000  18000   0.0%
  05   03      75     75   0.0%    1500   1500   0.0%   18000  17996   0.0%
  05   04      75     75   0.0%    1500   1500   0.0%   18000  18000   0.0%
  05   06      75     75   0.0%    1500   1500   0.0%   18000  18000   0.0%

  06   01      75     75   0.0%    1500   1500   0.0%   18000  18000   0.0%
  06   02      75     75   0.0%    1500   1500   0.0%   18000  18000   0.0%
  06   03      75     75   0.0%    1500   1500   0.0%   18000  17997   0.0%
  06   04      75     75   0.0%    1500   1500   0.0%   18000  18000   0.0%
  06   05      75     75   0.0%    1500   1500   0.0%   18000  18000   0.0%
```

#### show cloud monitor dataplane

この新しい show コマンドは、最新のデータプレーンモニタ情報に対して次の出力を表示するために導入されました。

**show cloud monitor dataplane**

```
  Cards      15 Second Interval     5 Minute Interval    60 Minute Interval
 Src  Dst    Miss    Hit    Pct    Miss    Hit    Pct    Miss    Hit    Pct
 ---  ---  ------ ------ ------  ------ ------ ------  ------ ------ ------
  02   01       0    150   0.0%       0   3000   0.0%       0  36000   0.0%
  03   01       0    150   0.0%       0   3000   0.0%      -incomplete-
  04   01       0    151   0.0%       0   3000   0.0%       0  36000   0.0%
  05   01       0    151   0.0%       0   3000   0.0%       0  36001   0.0%
  06   01       0    150   0.0%       0   3000   0.0%       2  35998   0.0%

  01   02       0    150   0.0%       0   3000   0.0%       0  36000   0.0%
  03   02       0    150   0.0%       0   3000   0.0%      -incomplete-
  04   02       0    150   0.0%       0   3000   0.0%       0  36000   0.0%
  05   02       0    150   0.0%       0   3000   0.0%       0  36000   0.0%
  06   02       0    151   0.0%       0   3001   0.0%       1  35999   0.0%

  01   03       0    151   0.0%       0   3000   0.0%      -incomplete-
  02   03       0    151   0.0%       0   3000   0.0%      -incomplete-
  04   03       0    150   0.0%       0   3000   0.0%      -incomplete-
  05   03       0    150   0.0%       0   3000   0.0%      -incomplete-
  06   03       0    151   0.0%       0   3000   0.0%      -incomplete-

  01   04       0    150   0.0%       0   3001   0.0%       0  36001   0.0%
  02   04       0    150   0.0%       0   3000   0.0%       0  36000   0.0%
  03   04       0    150   0.0%       0   3000   0.0%      -incomplete-
  05   04       1    149   0.7%       1   2999   0.0%       0  36001   0.0%
  06   04       0    150   0.0%       0   3000   0.0%       2  35998   0.0%

  01   05       1    149   0.7%       1   2999   0.0%       0  36000   0.0%
  02   05       0    150   0.0%       0   3000   0.0%       0  36000   0.0%
  03   05       0    150   0.0%       0   3000   0.0%      -incomplete-
  04   05       0    150   0.0%       1   2999   0.0%       1  35999   0.0%
  06   05       0    150   0.0%       0   3000   0.0%       2  35998   0.0%

  01   06       0    150   0.0%       0   3001   0.0%       0  36001   0.0%
  02   06       0    151   0.0%       0   3000   0.0%       1  35999   0.0%
  03   06       0    150   0.0%       0   3001   0.0%      -incomplete-
  04   06       0    150   0.0%       0   3000   0.0%       0  36000   0.0%
  05   06       0    150   0.0%       0   3000   0.0%       0  36000   0.0%
```

## バルク統計情報

この機能のサポートには、次の統計情報が含まれています。

### mon-di-net スキーマ

VPC-DI ネットワーク機能のモニタをサポートするため、次のバルク統計情報が mon-di-net スキーマに追加されています。

| バルク統計情報 | 説明 |
|---|---|
| cp-loss-5minave | 過去 5 分間の平均コントロールプレーン損失を示します。 |

| バルク統計情報 | 説明 |
|---|---|
| cp-loss-60minave | 過去 60 分間の平均コントロールプレーン損失を示します。 |
| dp-loss-5minave | 過去 5 分間の平均データプレーン損失を示します。 |
| dp-loss-60minave | 過去 60 分間の平均データプレーン損失を示します。 |

# DI ネットワークハートビートのしきい値の設定

次の手順では、損失したハートビートの割合が設定されたレベルを超えた場合に、SNMP アラームを生成するようにしきい値レベルを設定する方法について説明します。

（注） 内部高可用性タスク（HAT）は、内部 DI ネットワーク上の VM 全体のハートビートを常にモニタしています。この情報は、**show cloud monitor di-network summary** Exec モードコマンドを使用していつでも表示できます。

```
configure
  monitoring hat-5min-loss

  threshold hat-hb-5min-loss high_thresh [ clear low_thresh ]
default threshold hat-hb-5min-loss

  [ default ] threshold poll hat-hb-5min-loss interval duration

configure
  monitoring hat-60min-loss

  threshold hat-hb-60min-loss high_thresh [ clear low_thresh ]
default threshold hat-hb-60min-loss

  [ default ] threshold poll hat-hb-5min-loss interval duration
```

（注） この機能に関する補足情報については、『*Command Line Reference*』の「*Global Configuration Mode Commands*」の項を参照してください。

これらのしきい値を超過すると、次のアラームまたはトラップが生成されます。

- ThreshHatHb5MinLoss / ThreshClearHatHb5MinLoss
- ThreshHatHb60MinLoss / ThreshClearHatHb60MinLoss

これらのアラームまたはトラップの詳細については、『*SNMP MIB Reference*』を参照してください。

# ハートビート値の設定サポート

## 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 | • ASR 5500<br>• 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 | • *ASR 5500 System Administration Guide*<br>• *Command Line Interface Reference*<br>• *VPC-DI* システム管理ガイド<br>• *Statistics and Counters Reference* |

### マニュアルの変更履歴

| 改訂の詳細 | リリース |
|---|---|
| このリリースでは、管理カードとデータ処理カード間のデフォルトのハートビート値を変更し、管理カードがパケット処理カードを誤って検出して故障として報告しないようにできます。 | 21.8 |
| 最初の導入。 | 21.2 よりも前 |

## 変更された機能

特定の展開シナリオでは、管理カードは、約 2 秒間ハートビートを検出できない場合に、パケット処理カードを障害として報告します。この想定される障害は、内部 DI ネットワークの輻輳が原因でハートビートが遅延または損失したときに発生します。

このリリースでは、この問題に対処しています。

**以前の動作**：管理カードはデフォルト値の 2 秒以内にハートビートを検出できないため、パケット処理カードを故障として報告します。これにより、予定外のスイッチオーバーが発生します。

**新しい動作**：管理カードがパケット処理カードを誤って検出して故障として報告しないようにするために、管理カードとデータ処理カード間のデフォルトのハートビート値を変更できるようになりました。

**お客様への影響**：管理カードによるデータ処理の誤った報告、および予定外のスイッチオーバーを防ぎます。

## コマンドの変更

### high-availability fault-detection

上記の CLI コマンドは、**card hb-loss** *value* キーワードを含めるように拡張されています。このキーワードは、管理とパケット処理カード間のハートビート値の設定に使用されます。このコマンドはグローバル コンフィギュレーション モードで設定します。

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

**注**：

- **default**：ハートビート値をデフォルト値の 2 ハートビートに復元します。
- **card**：パケット処理カードを指定します。
- **hb-loss** *value*：ハートビート損失値を設定します。デフォルト値は 2 ハートビートです。
- 管理カードと管理カードの間のハートビート値は、デフォルト値の 2 ハートビートに設定されます。
- このコマンドは、管理カードとパケット処理カードの間のハートビート値のみを変更します。
- デフォルトでは、この CLI は無効になっています。

## 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、この機能のサポートにおける show コマンドまたはその出力について説明します。

### show heartbeat statistics hb-loss all

この show コマンドには、すべてのパケット処理カード用に次の新しいフィールドの値が含まれるようになりました。

- Max Bounces
- Total HB Miss
- Total HB Card Failure
  - Card/Cpu
  - Total
  - Age/Intf/Seqno/TimeStamp

- AFD（最も古いものから順に）

### show heartbeat statistics hb-loss card <card-number>

この show コマンドには、指定されたパケット処理カード用に次の新しいフィールドの値が含まれるようになりました。

- Max Bounces
- Total HB Miss
- Total HB Card Failure
  - Card/Cpu
  - Total
  - Age/Intf/Seqno/TimeStamp
  - AFD（最も古いものから順に）

# SF のモニタリング

アクティブ SF とスタンバイ SF ごとに NPU 統計情報を表示するには、**show npu utilization table** コマンドを使用します。過去 5 秒間、過去 5 分間、過去 15 分間の統計情報が報告されます。出力例を次に示します。

```
[local]swch91# show npu utilization table

 ******** show npu utilization table card 4 *******
                5-Sec Avg:
lcore00|lcore01|lcore02|lcore03|lcore04|lcore05|lcore06|lcore07|
                    IDLE:          |    99%|       |       |       |       |       |
     |
           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:          |       |       |       |       |       |       |
     |
          QUEUE_MCDMA_TX:          |       |       |       |       |       |       |
     |
      QUEUE_THREAD_MCDMA:          |       |       |       |       |       |       |
     |
       QUEUE_THREAD_VNPU:          |       |       |       |       |       |       |
```

```
|
     QUEUE_CRYPTO_RX:          |       |       |       |       |       |       |
|
    QUEUE_CRYPTO_IPC:          |       |       |       |       |       |       |
|
    QUEUE_THREAD_IPC:          |       |       |       |       |       |       |
|
         MCDMA_FLUSH:          |       |       |       |       |       |       |
|
QUEUE_THREAD_TYPE_MAX:         |       |       |       |       |       |       |
|
         300-Sec Avg:
lcore00|lcore01|lcore02|lcore03|lcore04|lcore05|lcore06|lcore07|
                IDLE:          |   99%|       |       |       |       |       |
|
       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:          |       |       |       |       |       |       |
|
      QUEUE_MCDMA_TX:          |       |       |       |       |       |       |
|
  QUEUE_THREAD_MCDMA:          |       |       |       |       |       |       |
|
   QUEUE_THREAD_VNPU:          |       |       |       |       |       |       |
|
     QUEUE_CRYPTO_RX:          |       |       |       |       |       |       |
|
    QUEUE_CRYPTO_IPC:          |       |       |       |       |       |       |
|
    QUEUE_THREAD_IPC:          |       |       |       |       |       |       |
|
         MCDMA_FLUSH:          |       |       |       |       |       |       |
|
QUEUE_THREAD_TYPE_MAX:         |       |       |       |       |       |       |
|
         900-Sec Avg:
lcore00|lcore01|lcore02|lcore03|lcore04|lcore05|lcore06|lcore07|
                IDLE:          |   99%|       |       |       |       |       |
|
       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:          |       |       |       |       |       |       |
      |
           QUEUE_MCDMA_TX:          |       |       |       |       |       |       |
      |
         QUEUE_THREAD_MCDMA:        |       |       |       |       |       |       |
      |
          QUEUE_THREAD_VNPU:        |       |       |       |       |       |       |
      |
            QUEUE_CRYPTO_RX:        |       |       |       |       |       |       |
      |
           QUEUE_CRYPTO_IPC:        |       |       |       |       |       |       |
      |
           QUEUE_THREAD_IPC:        |       |       |       |       |       |       |
      |
                MCDMA_FLUSH:        |       |       |       |       |       |       |
      |
     QUEUE_THREAD_TYPE_MAX:         |       |       |       |       |       |       |
      |


thread 1 IDLE                             99.32 %
thread 1 QUEUE_KNI_RX                      0.63 %
thread 1 QUEUE_PORT_RX                     0.05 %
-----------------------------------------------

 ******** show npu utilization table card 5 *******
                5-Sec Avg:
lcore00|lcore01|lcore02|lcore03|lcore04|lcore05|lcore06|lcore07|
                     IDLE:          |    99%|       |       |       |       |       |
      |
            QUEUE_PORT_RX:          |       |       |       |       |       |       |
      |
            QUEUE_PORT_TX:          |       |       |       |       |       |       |
      |
            QUEUE_VNPU_RX:          |       |       |       |       |       |       |
      |
            QUEUE_VNPU_TX:          |       |       |       |       |       |       |
      |
             QUEUE_KNI_RX:          |     0%|       |       |       |       |       |
      |
             QUEUE_KNI_TX:          |       |       |       |       |       |       |
      |
          QUEUE_THREAD_KNI:         |       |       |       |       |       |       |
      |
            QUEUE_MCDMA_RX:         |       |       |       |       |       |       |
      |
            QUEUE_MCDMA_TX:         |       |       |       |       |       |       |
      |
         QUEUE_THREAD_MCDMA:        |       |       |       |       |       |       |
      |
          QUEUE_THREAD_VNPU:        |       |       |       |       |       |       |
      |
            QUEUE_CRYPTO_RX:        |       |       |       |       |       |       |
      |
           QUEUE_CRYPTO_IPC:        |       |       |       |       |       |       |
      |
           QUEUE_THREAD_IPC:        |       |       |       |       |       |       |
      |
                MCDMA_FLUSH:        |       |       |       |       |       |       |
      |
     QUEUE_THREAD_TYPE_MAX:         |       |       |       |       |       |       |
      |
```

```
              300-Sec Avg:
lcore00|lcore01|lcore02|lcore03|lcore04|lcore05|lcore06|lcore07|
                     IDLE:          |    99%|       |       |       |       |       |
     |
            QUEUE_PORT_RX:          |       |       |       |       |       |       |
     |
            QUEUE_PORT_TX:          |       |       |       |       |       |       |
     |
            QUEUE_VNPU_RX:          |       |       |       |       |       |       |
     |
            QUEUE_VNPU_TX:          |       |       |       |       |       |       |
     |
             QUEUE_KNI_RX:          |     0%|       |       |       |       |       |
     |
             QUEUE_KNI_TX:          |       |       |       |       |       |       |
     |
          QUEUE_THREAD_KNI:         |       |       |       |       |       |       |
     |
            QUEUE_MCDMA_RX:         |       |       |       |       |       |       |
     |
            QUEUE_MCDMA_TX:         |       |       |       |       |       |       |
     |
        QUEUE_THREAD_MCDMA:         |       |       |       |       |       |       |
     |
         QUEUE_THREAD_VNPU:         |       |       |       |       |       |       |
     |
           QUEUE_CRYPTO_RX:         |       |       |       |       |       |       |
     |
          QUEUE_CRYPTO_IPC:         |       |       |       |       |       |       |
     |
          QUEUE_THREAD_IPC:         |       |       |       |       |       |       |
     |
               MCDMA_FLUSH:         |       |       |       |       |       |       |
     |
      QUEUE_THREAD_TYPE_MAX:        |       |       |       |       |       |       |
      |
              900-Sec Avg:
lcore00|lcore01|lcore02|lcore03|lcore04|lcore05|lcore06|lcore07|
                     IDLE:          |    99%|       |       |       |       |       |
     |
            QUEUE_PORT_RX:          |       |       |       |       |       |       |
     |
            QUEUE_PORT_TX:          |       |       |       |       |       |       |
     |
            QUEUE_VNPU_RX:          |       |       |       |       |       |       |
     |
            QUEUE_VNPU_TX:          |       |       |       |       |       |       |
     |
             QUEUE_KNI_RX:          |     0%|       |       |       |       |       |
     |
             QUEUE_KNI_TX:          |       |       |       |       |       |       |
     |
          QUEUE_THREAD_KNI:         |       |       |       |       |       |       |
     |
            QUEUE_MCDMA_RX:         |       |       |       |       |       |       |
     |
            QUEUE_MCDMA_TX:         |       |       |       |       |       |       |
     |
        QUEUE_THREAD_MCDMA:         |       |       |       |       |       |       |
     |
         QUEUE_THREAD_VNPU:         |       |       |       |       |       |       |
     |
           QUEUE_CRYPTO_RX:         |       |       |       |       |       |       |
     |
```

```
        QUEUE_CRYPTO_IPC:          |       |       |       |       |       |       |
    |
        QUEUE_THREAD_IPC:          |       |       |       |       |       |       |
    |
             MCDMA_FLUSH:          |       |       |       |       |       |       |
    |
    QUEUE_THREAD_TYPE_MAX:         |       |       |       |       |       |       |
     |

thread 1 IDLE                                99.37 %
thread 1 QUEUE_KNI_RX                         0.55 %
thread 1 QUEUE_PORT_RX                        0.08 %
-----------------------------------------------
```

**表 *11 : show npu utilization table***

| フィールド | 説明 |
|---|---|
| IDLE | 各コアのアイドル時間 |
| QUEUE_PORT_RX | RX ポートの処理にかかった時間 |
| QUEUE_PORT_TX | TX ポートの処理にかかった時間 |
| QUEUE_VNPU_RX | RX vNPU の処理にかかった時間 |
| QUEUE_VNPU_TX | TX vNPU の処理にかかった時間 |
| QUEUE_KNI_RX | RX カーネル ネットワーク インターフェイス（KNI）の処理にかかった時間。KNI は、IFTASK からのカーネルへのパスです。 |
| QUEUE_KNI_TX | TX KNI の処理にかかった時間 |
| QUEUE_THREAD_KNI | KNI 処理専用のスレッド |
| QUEUE_MCDMA_RX | RX マルチチャネル ダイレクト メモリアクセス（DMA）（MCDMA）の処理にかかった時間。MCDMA は、IFTASK から SESSMGR へのパスです。 |
| QUEUE_MCDMA_TX | TX MCDMA の処理にかかった時間。 |
| QUEUE_THREAD_MCDMA | MCDMA 処理専用のスレッド |
| QUEUE_THREAD_VNPU | VNPU 処理専用のスレッド |
| QUEUE_CRYPTO_RX | IPSec の処理にかかった時間 |
| QUEUE_CRYPTO_IPC | IPSec プロセス間通信（IPC）の処理にかかった時間 |
| MCDMA_FLUSH | MCDMA パケットのフラッシュにかかった時間 |
| QUEUE_THREAD_TYPE_MAX | 未使用 |

# 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。

第 15 章

# モニタプロセスのリスト



## 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 | • ASR 5500<br>• VPC-DI<br>• VPC-SI |
| 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 | • *ASR 5500 System Administration Guide*<br>• *Command Line Interface Reference*<br>• *Statistics and Counters Reference*<br>• *VPC-DI* システム管理ガイド<br>• *VPC-SI System* アドミニストレーション 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 |
|---|---|
| 最初の導入。 | 21.11 |

# 機能説明

モニタ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

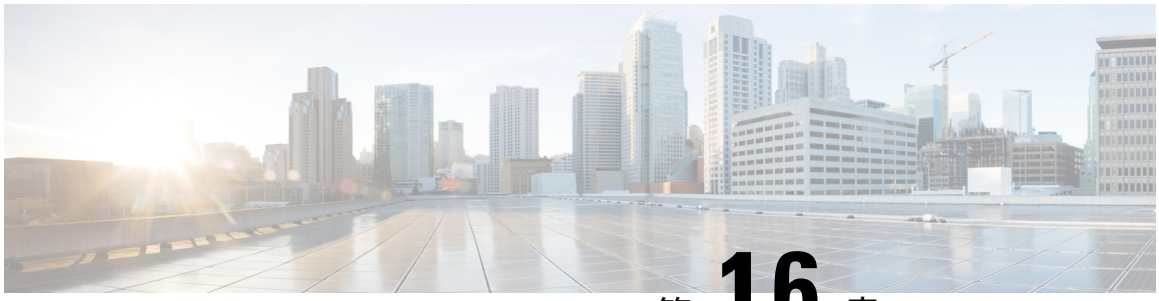
この項では、この機能のサポートにおける show コマンドやその出力に関する情報を提供します。

### 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。



第 16 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

**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<br>VPC-DI<br>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 | • *ASR 5500 System Administration Guide*<br>• *Command Line Interface Reference*<br>• *VPC-DI* システム管理ガイド<br>• *VPC-SI System* アドミニストレーション ガイド |

### 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 |
|---|---|
| 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。<br>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、SSD でのバルク統計情報サンプルの収集 （242 ページ） を参照してください。<br>この機能を有効にするため、**bulkstat** グローバル コンフィギュレーション モード コマンドに **config [ schema \| supplement ]** キーワードが追加されました。詳細については、個別のバルク統計情報設定ファイルの設定 （240 ページ） を参照してください。<br>**show configuration bulkstats brief** 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** リリース 20.0 以降の信頼できる StarOS のビルドでは、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 （238 ページ） およびオプションの設定 （239 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。

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（242 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（241 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（239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。

# 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*』のユーザドキュメント（リリース 20.0 より前に発行）で定義され、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、リリース 20.0 以降に発行された『*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 *12*：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 |

| 変数 | 説明 | 統計タイプ | データ タイプ |
|---|---|---|---|
| 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 |
| 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 |

| 変数 | 説明 | 統計タイプ | データ タイプ |
|---|---|---|---|
| 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 |
| localtzoffset | 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 |

| 変数 | 説明 | 統計タイプ | データ タイプ |
|---|---|---|---|
| swbuild | StarOS バージョンのビルド番号。 | 情報 | 文字列 |

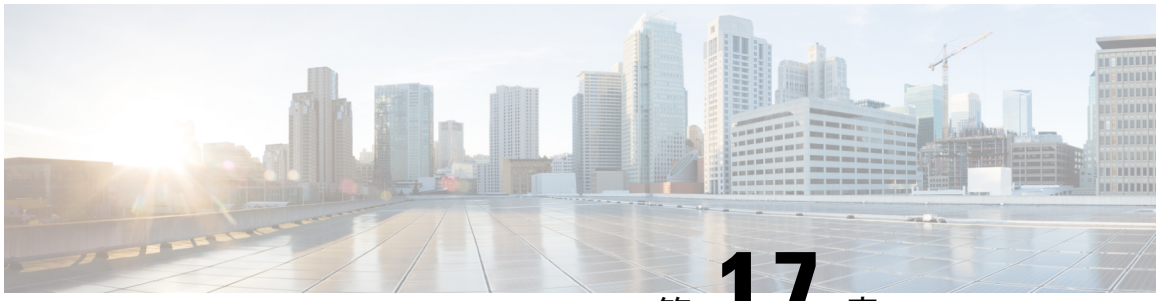
# バルク統計情報のイベントログメッセージ

stat ロギ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 *13 :* バルク統計情報に関連するイベントのロギング**

| イベント | イベント ID | 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 *filename* for storing bulkstats data」 |
| 受信者のオープンエラー | 31018 | 警告 | 「Unable to open url *filename* for storing bulkstats data」 |
| 受信者の書き込みエラー | 31019 | 警告 | 「Unable to write to url *filename* while storing bulkstats data」 |
| 受信側のクローズエラー | 31020 | 警告 | 「Unable to close url *filename* while storing bulkstats data」 |



第 17 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

**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<br>• UGP<br>• VPC-DI<br>• VPC-SI |

| | |
|---|---|
| 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 | • *ASR 5500 System Administration Guide*<br>• *Command Line Interface Reference*<br>• *VPC-DI* システム管理ガイド<br>• *VPC-SI System* アドミニストレーション ガイド |

### 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 |
|---|---|
| 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 サーバの IP アドレスを設定することもできます。<br>（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。 | 21.6 |
| 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、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 （255 ページ） を参照してください。<br>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。<br>**show snmp trap statistics** 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。 | 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、またはユーザ名に対して実行できます。
- **モニタ**：モニタロギングは、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。モニタは、サブスクライバの MSID またはユーザ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 サーバにログを送信するようにシステムを設定できます。

**重要** *15.0 MR4* より後のリリースの場合、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（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 verboseness 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、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パケット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*：パケットデータユニットのロギングに使用する verboseness 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ギン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ギングを無効にします。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ギングを有効にします。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ギン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 （258 ページ）とイベントの重大度 （292 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、ロギン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 （258 ページ）とイベントの重大度 （292 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 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**：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 サーバに送信できます。

**重要**

syslog サーバ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（258 ページ）とイベントの重大度（292 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）で使用されるプロコトル用 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ティングサービス（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 Proclet（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 Proclet 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）アプリケーション ファシリティ（リリース 20 の HENB-GW にはこのキーワードを使用しないでください）
- **henbgw**：HENB-GW ファシリティ（リリース 20 の HENB-GW にはこのキーワードを使用しないでください）
- **henbgw-pws**：HENB-GW パブリック警告システムのロギングファシリティ（リリース 20 の HENB-GW にはこのキーワードを使用しないでください）
- **henbgw-sctp-acs**：HENB-GW アクセス Stream Control Transmission Protocol（SCTP）ファシリティ（リリース 20 の HENB-GW にはこのキーワードを使用しないでください）
- **henbgw-sctp-nw**：HENBGW ネットワーク SCTP ファシリティ（リリース 20 の HNB-GW にはこのキーワードを使用しないでください）
- **henbgwdemux**：HENB-GW Demux ファシリティ（リリース 20 の HNB-GW にはこのキーワードを使用しないでください）
- **henbgwmgr**：HENB-GW マネージャファシリティ（リリース 20 の HNB-GW にはこのキーワードを使用しないでください）
- **hnb-gw**：HNB-GW（3G フェムト GW）のロギングファシリティ（リリース 20 の HNB-GW にはこのキーワードを使用しないでください）
- **hnbmgr**：HNB-GW Demux マネージャのロギングファシリティ（リリース 20 の HNB-GW にはこのキーワードを使用しないでください）
- **hss-peer-service**：ホームサブスクライバサーバ（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）アプリケーションのロギングファシリティ（このオプションは、このリリースではサポートされていません）。
- **mseg: gtpc**：MSEG Gtpc-C アプリケーションのロギングファシリティ（このオプションは、このリリースではサポートされていません）。
- **mseg-gtpu**：MSEG GTP-U アプリケーションのロギングファシリティ（このオプションは、このリリースではサポートされていません）。
- **msegmgr**：MSEG Demux マネージャのロギングファシリティ（このオプションは、このリリースではサポートされていません）。
- **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-rri**：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
- **ntfy-intf**：Notification Interface のロギングファシリティ [リリース12.0 以前のバージョンのみ]
- **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
- **proclet-map-frwk**：Proclet 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
- **rct**：リカバリ制御タスクのロギングファシリティ
- **rdt**：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-gtpc**：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
- **srdb**：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**：システムのロギングファシリティ
- **tacacsplus**：TACACS+ プロトコル ロギング ファシリティ
- **tcap**：TCAP プロトコルのロギングファシリティ
- **testctrl**：テストコントローラのロギングファシリティ
- **testmgr**：テストマネージャのロギングファシリティ
- **threshold**：しきい値のロギングファシリティ
- **ttg**：Tunnel Termination Gateway（TTG）のロギングファシリティ
- **tucl**：TCP/UDP コンバージェンスレイヤ（TUCL）のロギングファシリティ
- **udr**：ユーザデータレコード（UDR）ファシリティ（Charging サービスで使用）
- **user-data**：ユーザデータのロギングファシリティ
- **user-l3tunnel**：ユーザレイヤ 3 トンネルのロギングファシリティ
- **usertcp-stack**：ユーザ TCP スタック
- **vim**：Voice インスタントメッセージ（VIM）のロギングファシリティ
- **vinfo**：VINFO のロギングファシリティ
- **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ギングファシリティ
- **vpp**：Vector Packet Processing（VPP）のロギングファシリティ
- **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ギング ファシリティ
- **x2gw-demux**：X2GW demux タスクのロギングファシリティ

# トレースロギングの設定

トレースロギングは、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 }
```

# モニタログの設定

モニタロギングは、特定のサブスクライ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している場合に使用できます。

モニタは、サブスクライバのMSIDまたはユーザ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にのみ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。

## モニタログの有効化

モニタ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。

# モニタログの無効化

モニタ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 *14 :* 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 |
|---|---|
| **General Logging Statistics** | |
| 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。 |
| **Logging Source Statistics** | |
| 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）。 |
| **Runtime Logging Buffer Statistics** | |

| フィールド | 説明 |
|---|---|
| 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 サーバから、次のようにします。**システムが syslog サーバ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を指定する Universal Resource Locator（URL）を使用して設定されている場合にのみ、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**：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** リリース 20.0 以降の信頼できる StarOS のビルドでは、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
- ラストバウンス：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メッセンジャキューがいっぱいの場合は）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に通知します。

新しいthresholdコマンドを使用すると、ユーザ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ティング

チェックポインティングは、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。チェックポインティングを使用すると、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ティングを実行し、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ティングでは、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ティ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ティングによって、以前にチェックポインティ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ティングされたデータに置き換えられます。チェックポインティング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ティングを定期的に実行する必要があります。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** インスペクタレベルの管理ユーザは、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_verboseness* ] [ **facility** *facility* ] [**level** *severity_level* ] [ **pdu-data** *pdu_format* ] [ **pdu-verbosity** *pdu_verboseness* ] [ **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 *15 :* 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 |
| a11mgr | A11 マネージャファシリティ | 9000 ～ 9999 |
| 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 |

| ファシリティ | 説明 | イベント ID の範囲 |
|---|---|---|
| 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 |
| cap | CAMEL Application Part（CAP）ファシリティ | 87900 ～ 88099 |
| chatconf | CHATCONF ファシリティ | 74000 ～ 74999 |

| ファシリティ | 説明 | イベント ID の範囲 |
|---|---|---|
| cli | CLI（コマンド ライン インターフェイス）のロギング ファシリティ | 30000 ～ 30999 |
| connproxy | 接続プロキシファシリティ | 190000 ～ 190999 |
| 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 ～ 92849 |
| 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 |

| ファシリティ | 説明 | イベント ID の範囲 |
|---|---|---|
| diabase | Diabase メッセージファシリティ | 92800 ～ 92809 |
| 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 ～ 121999 |
| 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 |
| 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 |

| ファシリティ | 説明 | イベント ID の範囲 |
|---|---|---|
| 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 |
| gtpp | 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 の範囲 |
|---|---|---|
| gy-diameter | Gy Diameter メッセージファシリティ | 92810 ～ 92819 |
| 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 ～ 34999 |
| 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 ～ 132999 |
| hddshare | HDD 共有ファシリティ | 184000 ～ 184999 |
| henb-gw | Home eNodeB-GW ファシリティ | 195000 ～ 195999 |
| henbapp | Home eNodeB アプリケーションファシリティ | 196000 ～ 196999 |
| henbgwdemux | Home eNodeB-GW Demux ファシリティ | 194000 ～ 194999 |
| henbgwmgr | Home eNodeB-GW マネージャファシリティ | 193000、193999 |
| hnb-gw | Home NodeB（HNB）ゲートウェイファシリティ | 151000 ～ 151999 |
| hnbmgr | HNB マネージャファシリティ | 158000 ～ 158199 |
| hss-peer-service | ホーム サブスクライバ サーバ（HSS）ファシリティ [MME] | 138000 ～ 138999 |
| igmp | Internet Group Management Protocol（IGMP） | 113000 ～ 113999 |
| ikev2 | IKEv2 ファシリティ | 122000 ～ 122999 |
| ims-authorizatn | IMS 承認サービス ライブラリファシリティ | 98100 ～ 98999 |
| ims-sh | IMS SH ライブラリファシリティ | 124000 ～ 124999 |
| imsimgr | International Mobile Subscriber Identity（IMSI）マネージャファシリティ | 114000 ～ 114999 |
| imsue | IMS User Equipment（IMSUE）ファシリティ | 144000 ～ 145999 |

| ファシリティ | 説明 | イベント ID の範囲 |
|---|---|---|
| ip-arp | IP Address Resolution Protocol（ARP）ファシリティ | 19000 ～ 19999 |
| ip-interface | IP インターフェイスファシリティ | 18000 ～ 18999 |
| ip-route | IP ルートファシリティ | 20000 ～ 20999 |
| 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 ～ 134999 |
| ipne | IP Network ENABLER（IPNE）ファシリティ | 192000 ～ 192999 |
| ipsec | IPSecプロトコルファシリティ | 55000 ～ 56998 |
| ipsg | IP サービスゲートウェイ（IPSG）ファシリティ | 128000 ～ 128999 |
| ipsgmgr | IPSG マネージャ（IPSGMgr）ファシリティ | 99000 ～ 99999 |
| ipsp | IP プール共有プロトコル（IPSP）ファシリティ | 68000 ～ 68999 |
| 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 |

| ファシリティ | 説明 | イベント ID の範囲 |
|---|---|---|
| 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 |

| ファシリティ | 説明 | イベント ID の範囲 |
|---|---|---|
| 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 |
| mseg-app | Mobile Services Edge Gateway（MSEG）アプリケーション ファシリティ<br>本リリースではサポートされていません。 | 172300 ～ 172999 |
| mseg-gtpc | MSEG GTPC アプリケーション ファシリティ<br>本リリースではサポートされていません。 | 172000 ～ 172199 |
| mseg-gtpu | MSEG GTPU アプリケーション ファシリティ<br>本リリースではサポートされていません。 | 172200 ～ 172299 |
| msegmgr | MSEG マネージャファシリティ<br>本リリースではサポートされていません。 | 171000 ～ 171999 |
| mtp2 | Message Transfer Part 2（MTP2）サービスファシリティ [SS7] | 116900 ～ 116999 |
| mtp3 | Message Transfer Part 3（MTP3）サービスファシリティ [SS7] | 115600 ～ 115699 |
| multicast-proxy | マルチキャスト プロキシ ファシリティ | 94000 ～ 94999 |
| 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 ～ 153999 |
| netwstrg | ネットワーク ストレージ ファシリティ | 78000 ～ 78999 |
| npuctrl | ネットワーク処理ユニット（NPU）制御ファシリティ | 16000 ～ 16999 |
| npudrv | NPU ドライバファシリティ | 191000 ～ 191999 |

| ファシリティ | 説明 | イベント ID の範囲 |
|---|---|---|
| npumgr | NPU マネージャ（NPUMGR）ファシリティ | 17000 ～ 17999 |
| npumgr-acl | NPUMGR ACL ファシリティ | 169000 ～ 169999 |
| npumgr-drv | NPUMGR ドライバファシリティ | 185000 ～ 185999 |
| npumgr-flow | NPUMGR フローファシリティ | 167000 ～ 167999 |
| npumgr-fwd | NPUMGR 転送ファシリティ | 168000 ～ 168999 |
| npumgr-init | NPUMGR 初期化ファシリティ | 164000 ～ 164999 |
| npumgr-lc | NPUMGR LC ファシリティ | 180000 ～ 180999 |
| npumgr-port | NPUMGR ポートファシリティ | 166000 ～ 166999 |
| npumgr-recovery | NPUMGR リカバリファシリティ | 165000 ～ 165999 |
| npumgr-vpn | NPUMGR VPN ファシリティ | 181000 ～ 181999 |
| npusim | NPUSIM ファシリティ | 176000 ～ 176999 |
| ntfy-intf | イベント通知インターフェイス ファシリティ | 170000 ～ 170499 |
| orbs | orbs：オブジェク リクエストブローカ（ORB）システムファシリティ | 15000 ～ 15999 |
| 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 |

| ファシリティ | 説明 | イベント ID の範囲 |
|---|---|---|
| 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 |

| ファシリティ | 説明 | イベント ID の範囲 |
|---|---|---|
| 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 ～ 152009 |
| s1ap | 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 |
| 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 ～ 155199 |
| 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 ～ 173199 |
| 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 ～ 116599 |

| ファシリティ | 説明 | イベント ID の範囲 |
|---|---|---|
| sgsn-gtpu | SGSN GTP-U プロトコルファシリティ | 86900 ～ 87099 |
| sgsn-mbms-bearer | SGSN MBMS ベアラーアプリケーション（SMGR）ファシリティ | 116600 ～ 116799 |
| 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 ～ 117999 |
| 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 ～ 92859 |
| 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 ～ 159999 |
| 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 の範囲 |
|---|---|---|
| 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 ～ 174999 |
| testmgr | テスト マネージャ ファシリティ | 175000 ～ 175999 |
| 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 ～ 130999 |
| 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 | ユーザデータレコード（UDR）ファシリティ | 79000 ～ 79999 |
| user-data | ユーザデータファシリティ | 51000 ～ 51999 |
| user-l3tunnel | ユーザ L3 トンネルファシリティ | 75000 ～ 75999 |
| usertcp-stack | ユーザ TCP スタックファシリティ | 173300 ～ 173499 |

| ファシリティ | 説明 | イベント ID の範囲 |
|---|---|---|
| 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 ～ 104999 |
| wimax-r6 | WiMAX R6 プロトコル（シグナリング）ファシリティ | 104000 ～ 104899 |

# イベントの重大度

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、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、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、重大度レベルが高いイベントもログに記録します。
- **情報**：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重大度を持つイベントをログに記録します。
- **デバッグ**：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「重大度」レベルに対応します。したがって、「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。

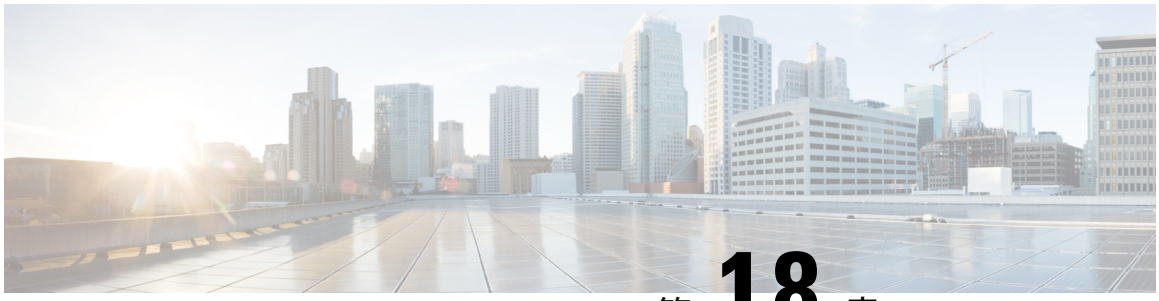
次に、ログに記録されたイベントの出力例を示します。

```
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 *16 :* 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 |
| [cli 30005 info] | 以下を含むイベントに関する情報<br>• イベントが属している機能<br>• イベント ID<br>• イベントの重大度<br>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、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 18 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

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ト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。

## 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 192.168.250.1 (192.168.250.1): 56 data bytes
64 bytes from 192.168.250.1: icmp_seq=0 ttl=255 time=0.4 ms
64 bytes from 192.168.250.1: icmp_seq=1 ttl=255 time=0.2 ms
64 bytes from 192.168.250.1: icmp_seq=2 ttl=255 time=0.2 ms
64 bytes from 192.168.250.1: icmp_seq=3 ttl=255 time=0.2 ms
64 bytes from 192.168.250.1: icmp_seq=4 ttl=255 time=0.2 ms
--- 192.168.250.1 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トを行います。

# 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_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、出力例を示します。

```
traceroute to 192.168.250.1 (192.168.250.1), 30 hops max, 40 byte packets
 1  192.168.250.1 (192.168.250.1)  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_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、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       SPIO1
*10.0.4.0/24      0.0.0.0     kernel      0       0       SPIO1
*10.0.4.0/32      0.0.0.0     kernel      0       0       SPIO1
*10.0.4.3/32      0.0.0.0     kernel      0       0       SPIO1
*10.0.4.255/32    0.0.0.0     kernel      0       0       SPIO1
```

# 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。

# プロトコルモニタの使用

プロトコルモニタには、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 （305 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。

## 特定サブスクライバのプロトコルモニタの使用

プロトコルモニタは、現在処理中の特定のサブスクラ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。

特定のサブスクラ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。

マルチコールト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、複数のプロトコル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ライバに対するモニタの出力例の一部を示します。デフォルトのプロトコルがモニタされました。

```
------------------------------------------------------------------------------
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**：システム上ので検出された 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コ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。

*20.0* より前のリリースでは、SSD がすでに実行されている間、オペレータは別の SSD を開始できました。大規模な設定では、最初の 1 つが完了するのを待つ間、SSD の要求がタイムアウトになることがよくあります。オペレータはタイムアウト全体が期限切れになるまで障害を認識していませんでした。オペレータは、同時に複数の SSD 実行させることができます。

*20.0* 以降のリリースでは、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、ユーザ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実行します。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もあります。

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。

# ハイパーバイザの強制再起動

ハイパーバイザは、仮想ウォッチドッグデバイスを提供するために必要です。基盤となるホスト OS の障害が原因で、StarOS がこのウォッチドッグデバイスのサービスを停止した場合、ハイパーバイザは VM を強制的に再起動する必要があります。

VPC-DI インスタンスである仮想化されたシャーシでは、ホストの障害に起因する別の VM の再起動を CF がリモートで行う方法はありません。正常な状態では、CF は VM 上で実行されている StarOS にメッセージを送信することで、別の VM をリモートで再起動できます。ただし、ホストが障害状態の場合、これは機能しない可能性があり、ハイパーバイザのウォッチドッグはフォールバックメカニズムとして機能する必要があります。

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 は、独自のウォッチドッグメカニズムを提供します。詳細については、VMware のドキュメントを参照してください。

**表 *17*：障害状況および再起動方法**

| 障害状況 | 再起動方法 | リカバリ | 注意 |
|---|---|---|---|
| **card reboot** *x* | CF は、リモート StarOS にローカルリブートを実行するように指示します。 | | |

| 障害状況 | 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 HATCPU タスクがウォッチドッグのサービスを停止します。 |
| カーネルのハング/クラッシュ | カーネルの再起動またはハイパーバイザのウォッチドッグ | 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 | CF は、リモート StarOS にローカルリブートを実行するように指示します。 | 参照先：VMware HA クラスタ。 |

# 手動によるスタンバイ CF へのスイッチング

手動によりアクティブCF「カード」を強制的に冗長ペアのスタンバイCFに切り替えることができます。

次の Exec モード CLI は、アクティブ CF をスタンバイ CF に切り替えます。

**card switch to** *slot*

注：

- このコマンドは、VPC-DI インスタンス内の冗長 CF VM に対してのみ機能します。
- スイッチオーバーを開始するには、アクティブ CF にログインする必要があります。
- スロット には 1 または 2 を指定できます。

スイッチオーバープロセスの流れは、次のとおりです。

1. アクティブ CF はスタンバイモードに移行します。
2. スタンバイ CF はアクティブモードに移行します。
3. リロードは、以前のアクティブ CF（現在はスタンバイ CF）で開始されます。

第 19 章

# パケットキャプチャ（PCAP）トレース



## 機能情報

**要約データ**

| | |
|---|---|
| 該当製品または機能エリア | • ePDG<br>• IPSec<br>• MME<br>• SaMOG |
| 該当プラットフォーム | ASR 5500<br>vPC-SI<br>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 | • *ASR 5000 System Administration Guide*<br>• *ASR 5500 System Administration Guide*<br>• *Command Line Interface Reference Guide*<br>• *ePDG Administration Guide*<br>• *IPSec Reference Guide*<br>• *SaMOG Administration Guide*<br>• *VPC-SI System* アドミニストレーション ガイド |

### 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、MME S1-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。

# 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、シャーシ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。

- **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、シャーシ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。

- **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（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**　ローテンション時間は 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場合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に含めるには、**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に割り込むことになります。
  - **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 *18* : ***show cdr statistics*** コマンド出力の説明

| フィールド | 説明 |
|---|---|
| **EDR-UDR ファイルの統計情報**： | |
| 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 レコード固有の統計情報**： | |
| 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。 |

| 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 *19 : 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
```

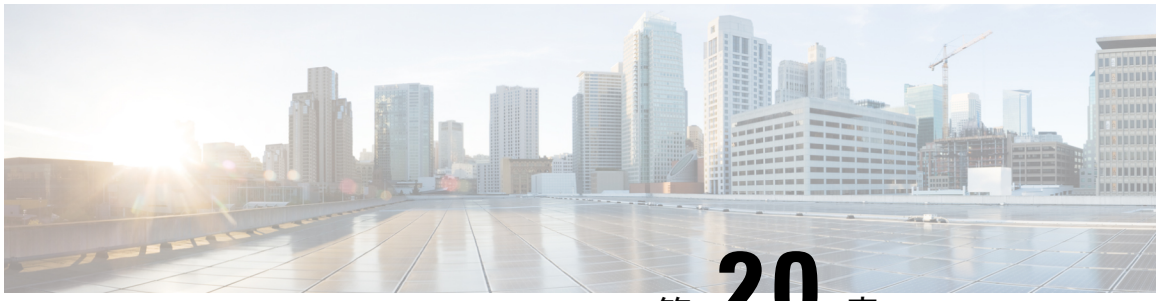
**重要** 16 進ダンプモジュールの統計情報をクリアし、リセットするには、Exec モードで **clear hexdump-module statistics** コマンドを使用します。

表 *20* : ***show hexdump-module statistics***コマンド出力の説明

| フィールド | 説明 |
|---|---|
| **Hexdump-module-Record file Statistics**： | |
| 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。 |
| **Hexdump-module PUSH Statistics**： | |
| 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。 |
| **Primary Server Statistics**： | |
| Successful File Transfers | プライマリストレージサーバ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に16進ダンプファイルを転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に16進ダンプファイルを転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に対する最後のプッシュ操作が完了した時刻。 |
| **Secondary Server Statistics**： | |
| Successful File Transfers | セカンダリストレージサーバ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に16進ダンプファイルを転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に16進ダンプファイルを転送するためのプッシュ操作に失敗した合計回数。 |

| フィールド | 説明 |
|---|---|
| Num of periodic PUSH | セカンダリストレージサーバ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に対する最後のプッシュ操作が完了した時刻。 |



第 20 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、サブスクライ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

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。

- *16.1* より前の *StarOS* リリースでは、イメージファイル名はリリースバージョンと対応するビルド番号によって識別されます。Format = **production.***build_number***.***platform***.bin**.
- *StarOS* リリース *16.1* 以降では、イメージファイル名はプラットフォームのタイプとリリース番号によって識別されます。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、サブスクライ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。

# ブートシーケンスの割り込み

「Booting priority」メッセージ行が表示されたとき（それより前ではない）、次の例に示すように、CTRL+C を押して起動プロセスを中断します。

```
Booting priority 8
   image : /flash/image_filename.bin
   config: /flash/system.cfg
Entry at 0x000000000cba45e0
```

シーケンスのこの時点で CTRL+C を押します。

起動プロセスが中断された後に、次のようなメッセージが表示されます。

```
*******9/0 Ctrl-C Pressed-----------------------------------------------------------
Failed.
   aborted by user
8/0:boot>
```

# 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 | /pcmcia1 | /usb1 }/***filename*

# 選択したイメージからの起動

システムリカバリプロセスを開始するには、ブート CLI で **boot** コマンドを発行します。

# 設定ファイルを使用しない起動

この手順では、設定ファイルをロードせずに、指定した起動イメージを使用してシステムを起動します。次に、サンプルコマンドの文字列を示します。

**8/0:cli>boot -noconfig /flash/***image_filename***.bin**

起動シーケンスは、設定ファイルを作成するためのクイック セットアップ ウィザードを開始するプロンプトで終了します。

```
Launching StarOS
Starting program at 0x0000000000100000
Starent Networks ASR5500 Intelligent Mobile Gateway
management_card is starting up.............................
Starting software image_version_number...
No configuration found, press enter to continue.
1. Do you wish to continue with the Quick Setup Wizard[yes/no]:
```

クイック セットアップ ウィザードを終了するには、上記のプロンプトの応答として **no** と入力します。次の例に示すように、Exec モードの **configure** コマンドを使用し、その後に設定ファイルの URL を指定することで、必要な設定ファイルをロードします。

[local]*host_name*# **configure /flash/system.cfg**

# 指定された設定ファイルを使用した起動

この手順では、指定された起動イメージとコンフィギュレーション ファイルを使用してシステムを起動します。次に、サンプルコマンドの文字列を示します。

**8/0:cli>boot -config=/flash/system.cfg /flash/***image_filename***.bin**

ブートシーケンスは、CLI プロンプトが表示された状態で終了します。

[local]*host_name*#

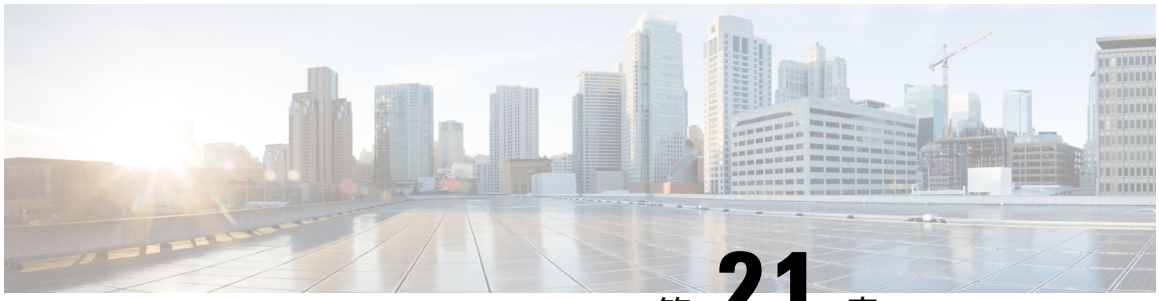
Exec モードの **show configuration** コマンドを実行して、必要な設定がロードされていることを確認します。

# 不正なスタートアップ設定ファイルからの復旧

スタートアップ コンフィギュレーション ファイルが破損している場合、または使用できない場合（たとえば、管理者または無効なパスワードが設定された空のコンフィギュレーション ファイルなど）は、次のように VPC VM を回復できます。

- VM を再起動します。
- SYSLINUX「boot:」プロンプトタイプ *priority_number* **config=** where *priority_number* は、既知の正常な .bin ファイルのブートの優先順位です。

VM はその優先順位の .bin ファイルで起動しますが、スタートアップ コンフィギュレーションはありません。



第 21 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

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 （326 ページ）
- ルールの順序 （328 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す。

## 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*（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* 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*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することができます。

- 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** 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 （329 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。サブスクライバトラフィックのアクションと基準の設定 （329 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 （330 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 （330 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。

# サブスクライバトラフィックのアクションと基準の設定

サブスクライ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 （328 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 （333 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 （335 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 （337 ページ）
- 複数のサブスクライバへの単一 ACL の適用 （341 ページ）
- 複数のサブスクライバへの単一 ACL の適用 （341 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 *6 : ACL* の処理順序**

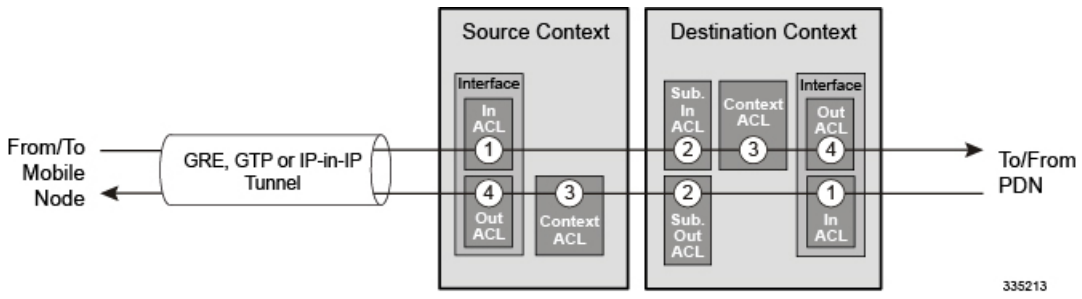



**表 *21 : ACL* の処理順序の説明**

| |
|---|
| 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 | サブスクライバに対して設定されたインバウンドACL（特定のサブスク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クライバのいずれか）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 | サブスクライバに対して設定されたアウトバウンドACL（特定のサブスク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クライバのいずれか）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の数が、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 （333 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 （334 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 がカーネル宛て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の数が、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** 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 （334 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 （335 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 ベースのサブスクライバにおける 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ライバに適用されます。サブスク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 ベースのサブスクライバに適用するには、**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、StarOS 12.3 またはそれ以前のリリースを使用している場合は、『*AAA and GTPP Interface Administration and Reference*』を参照してください。StarOS 14.0 以降のリリースを使用している場合は、『*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ライバに 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** 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。個々のサブスクライバへの ACL の適用 （337 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。個々のサブスクライバへの ACL 設定の確認 （337 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。

# 個々のサブスクライバへの ACL の適用

ACL を個々のサブスクライバに適用するには、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の数が、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。

## 個々のサブスクライバへの 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ライバ *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
gtpp group default
exit
content-filtering server-group cfsg_name
   response-timeout response_timeout
   connection retry-timeout retry_timeout
         end
```

# default というサブスクライバへの ACL の適用

この項では、*default* という名前のサブスクライバに 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** 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 というサブスクライバへの ACL の適用 （338 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 というサブスクライバに対する ACL 設定の確認 （339 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 というサブスクライバへの ACL の適用

*default* というサブスクライバに 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の数が、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 というサブスクライバに対する 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 というサブスクライ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
      gtpp group default
      exit
      content-filtering server-group cfsg_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ライバへの 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ライバへの 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** default というサブスクライバへの ACL の適用 （338 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ライバへの ACL 設定の確認 （340 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ライバへの ACLの適用

ACL をサービス指定のデフォルトサブスクライバに適用するには、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の数が、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ライバへの 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ライ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
```

# 複数のサブスクライバへの単一 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クライバに適用されます。サブスク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ク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ク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 *22*:「デフォルト」サブスクライバ属性を提供するために使用される機能

| 機能 | 説明 |
|---|---|
| *default* という名前のサブスクライバ | システムは各コンテキスト内に *default* というサブスクライバを作成します。*default* というサブスクライバのプロファイルには、そのコンテキストで認証されたサブスクライバの属性値の設定テンプレートが備わっています。<br>RADIUS ベースのサブスクライバプロファイルに含まれていないサブスクライバ属性は、*default* というサブスクライバに定義されている属性の値に従って設定されます。<br>**注**：*default* というサブスクライバのプロファイルを使用してローカルに設定されているサブスクライバの欠落情報を提供することはできません。 |
| **default subscriber** | 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の「default」サブスク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* というサブスク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クライバ。
- サブスク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クライバを「デフォルト」のプロファイルとして使用するように設定することで促進されたすべてのサブスクライバ。

## 複数のサブスクライバへの APN を介した ACL の適用

APN を介して複数のサブスクライバに 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**

## 複数のサブスクライバへの 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クライバに適用される場合、サブスク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 サブスクライバの 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ク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。

サブスク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** 複数のサブスクライバへの APN を介した ACL の適用 （342 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 （343 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：

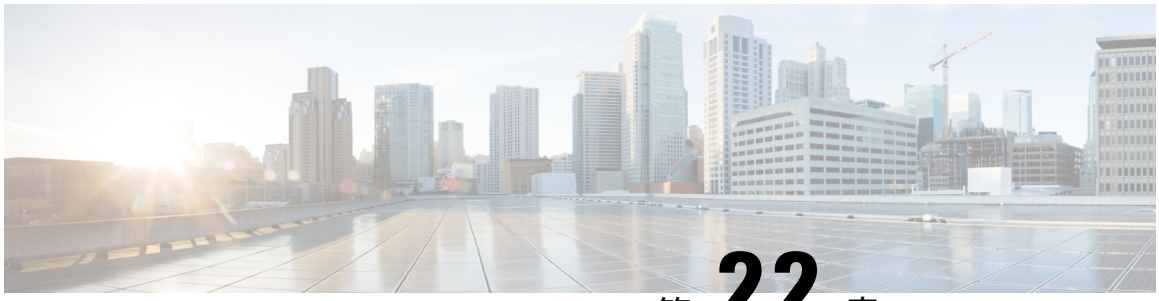
Exec モードで次のコマンドを入力して、ACL リストが適切に適用されていることを確認します。

**show configuration context** *context_name*

*context_name* は、ACL が適用されたデフォルトのサブスクライバを持つ 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 22 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

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 （346 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 （347 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 （348 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ライ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ライバの接続解除 （349 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ライバの接続解除

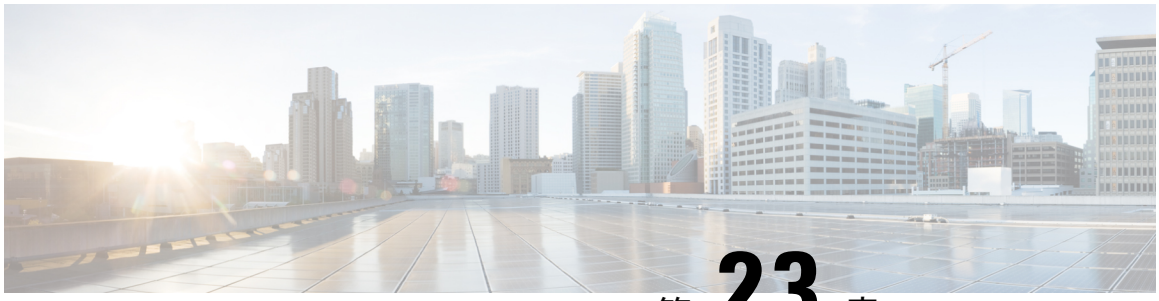
システム負荷が高いときは、許容レベルのシステムパフォーマンスを維持するために、サブスクライバの切断が必要になる場合があります。しきい値を設定して、コールが接続または非アクティブになっている時間に基づいて、コールを切断するサブスクライバを選択できます。

現在選択されているサブスクライ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
```

サブスクライ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
```



第 23 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

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の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 } reg_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 isp1
      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** VPC-DI 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している必要があります。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** VPC-DI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計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ディ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 VPC-DI 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 OSPFv6 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、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 ～ 10（*18.2* よりも前のリリース）または 1 ～ 32（リリース *18.2+*）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ドバ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　VPC-DI 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ドバタイズされます。BGP-4 プロトコルは、シャーシ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、「シャーシ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することです。BGP-4 は、サブスクライ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（ルーターゲット）

BGP 拡張コミュニティは、ルーターゲットを定義します。MPLS VPN は、ルーターゲット（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す。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 icsr-aggr-advertisement-interval** コマンドは、BGP アドレスファミリ（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（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 icsr-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 *23 : 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 |
|---|---|
| **accept-zero-as-rd** | 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。 |
| **address-family { ipv4 \| ipv6 }** | IPv4 または IPv6 アドレスファミリ設定モードを開始します。 |
| **address-family { vpnv4 \| vpnv6 }** | VPNv4 または VPNv6 アドレスファミリ設定モードを開始します。 |
| **bgp graceful-restart { restart-time** *rest_time* **\| stalepath-time** *stale_time* \| **update-delay** *delay* | 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。 |
| **description** *text* | この設定の説明テキストを入力できます。 |
| **distance { admin** *distance* **prefix** *prefix_addr* **[ route-access-list** *list_name* **] \| bgp external** *ebgp_dist* **internal** *ibgp_dist* local *local_dist* **}** | 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 |
|---|---|
| **enforce-first-as** | Exterior Border Gateway Protocol（eBGP）ルートの最初の AS を適用します。 |
| **ip vrf** *vrf_name* | 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。 |
| **maximum-paths { ebgp** *max_num* **\| ibgp** *max_num* **}** | 複数のパスを介したパケットの転送を有効にし、ネイバー間の外部 BGP（eBGP）パスまたは内部 BGP（iBGP）パスの最大数を指定します。 |
| **neighbor** *ip_address* **{ activate \| advertisement-interval** *adv_time* **\| capability graceful-restart \| default-originate [ route-map** *map_name* **] \| distribute-list** *dist_list***{ in \| out } \| ebgp-multihop [ max-hop number ] \| encrypted password** e*ncrypted_password* **\| fall-over bfd [ multihop ] \| filter-list filt_list { in \| out } \| max-prefix** *max_num* **[ threshold** *thresh_percent* **] [ warning-only ] \| next-hop-self \| password** *password* **\| remoteas** *AS_num* **\| remove-private-AS \| restart-time** *rest_time* **\| route-map** *map_name* **{ in \| out } \| send-community { both \| extended \| standard } \| shutdown \| srp-activated-soft-clear \| timers { [ connect-interval** *conn_time* **] \| [ keepalive-interval** *keep_time* **holdtimeinterval** *hold_time* **] } \| update-source** *ip_address* **\| weight** *value* **}** | 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ることに注意してください。<br><br>注：**advertisement-interval** は、そのアドレスファミリに有効になるように、アドレスファミリ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 アドレスファミリにのみ適用されます。**address-family** コマンド使用してアドレスファミリを指定します。その後で、アドレス ファミリ 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。 |
| **network** *ip_address/mask* **[ route-map** *map_name* **]** | BGP を介してアナウンスするネットワークを指定します。 |
| **redistribute { connected \| ospf \| rip \| static } [ route-map** *map_name* **]** | BGP を介して別のプロトコルから BGP にルートを再配布します。 |
| **router-id** *ip_address* | 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。 |
| **scan-time** *time* | 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トのネクストホップをモニタして、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。 |
| **timers bgp keepalive-interval** *interval* **holdtime-interval** *time* **[ min-peer-holdtimeinterval** *time* **]** | 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 | • *Command Line Interface Reference*<br>• *Statistics and Counters Reference*<br>• *VPC-DI* システム管理ガイド |

### 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

アーキテクチャを備えた Cisco 仮想化パケットコア分散型インスタンス（VPC-DI）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。
- **cost***number*：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。
- **cost***number*：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 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** VPC-DI 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、ASR 5500 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

```
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 （374 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 （375 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 （374 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 （375 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 （376 ページ）。
- ICSR Dead 間隔を無視する BFD の設定 （376 ページ）。
- ICSR スイッチオーバー ガード タイマー の設定 （377 ページ）。
- BFD マルチホップ フォールオーバーの有効化 （378 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（379 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 （375 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（365 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。

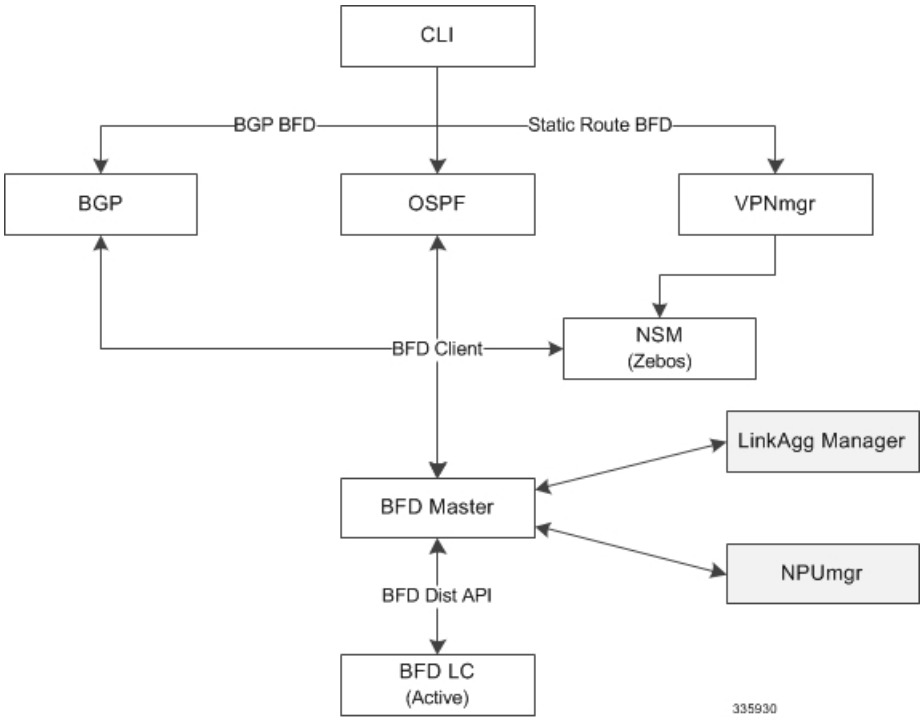
# リンク集約メンバーリンクの BFD サポート

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 *7: 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 の整数としてで指定します。
- **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。

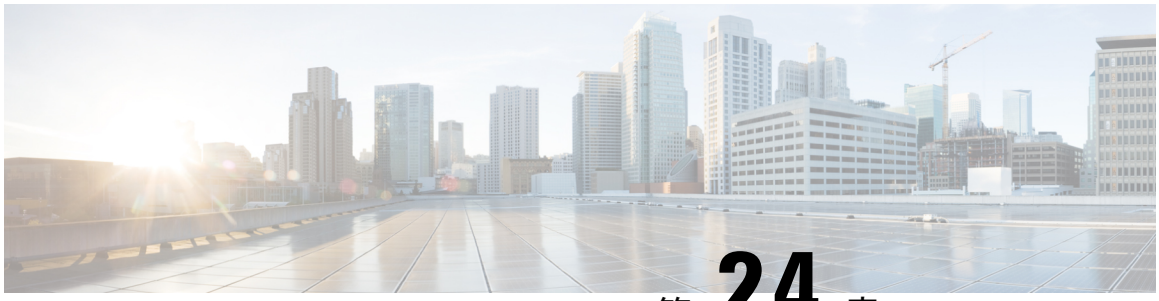
# ルーティング情報の表示

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 24 章

# VLAN

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。

VPC は VLAN を使用して、DI ネットワーク内でシグナリングサービスとベアラーサービスを実装します。



## 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て拡張でき、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。サブスクライバアドレスは、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 サーバグループの RADIUS NAS-IP アドレスのオーバーラッピング
- さまざまな RADIUS サーバグループの RADIUS サーバ 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ティングメッセージを、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、DI ネットワーク用の VLAN とそのゲートウェイ製品のいくつかをサポートしています。

（注） VLANは、サービスコンテキストのSF vNIC トラフィックポート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-DI 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 （386 ページ）
• VLAN および VMware （387 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VLAN ID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

- *KVM/Networking*：https://help.ubuntu.com/community/KVM/Networking

# VLAN および VMware

VMware は、ネットワーク展開入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キング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サブスクライバの関連付けを設定します。詳細については、サブスクライバ VLAN の関連付けの設定 （389 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 （390 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サブスクライバの関連付けを設定します。詳細については、サブスクライバ VLAN の関連付けの設定 （389 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 （390 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。

# サブスクライバ VLAN の関連付けの設定

サブスクライバトラフィックは、ユーザ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 を介してサブスクライ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、サブスクライ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 サーバ上のサブスクライバプロファイル内で設定して、特定の VLAN をサブスクライ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 サブスクライバプロファイルの設定手順はRADIUSサーバアプリケーション間で異なるため、この項では、サブスクライ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。サブスクライバの設定手順については、RADIUSサーバに付属のマニュアルを参照してください。

## ローカルサブスクライ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ライ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（388ページ）に記載されている手順に従って、サブスクライバタイプの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
```

## サブスクライバのプロファイル設定の確認

サブスクライ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：

- 各サブスクライ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。

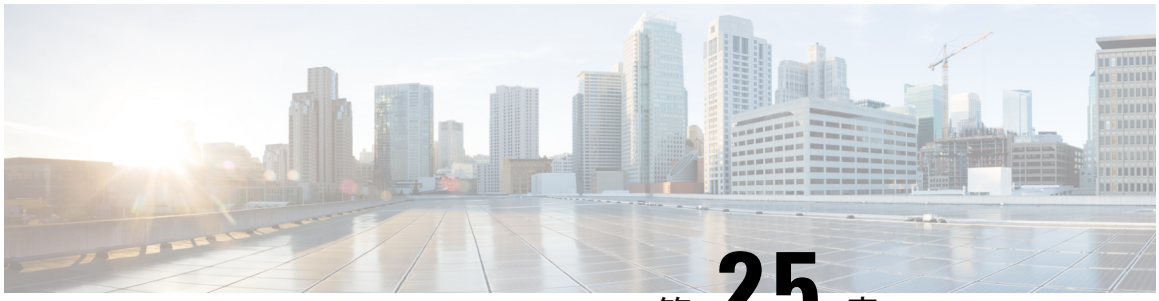
表 *24 : VLAN* 関連の設定コマンド

| CLI モード | コマンド | 説明 |
|---|---|---|
| AAA サーバ グループ コンフィギュレーション モード | **radius attribute nas-ip-address address** *ip_address* **nexthop-forwarding-address** *ip_address* **vlan** *vlan_id* | 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。<br>**注**：**vlan** 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 **aaa-large configuration** 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 | **ip vlan** *vlan_id* | 接続先コンテキストのサブスクライバトラフィックに関連付けるVLAN識別子を設定します。 |
| コンテキスト コンフィギュレーション モード | **ip pool** *pool_name* **nexthop forwarding address** *ip_address* **overlap vlanid** *vlan_id* | テクストホップ転送アドレスが設定されている場合、**overlap vlanid** 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 | **ip routing overlap-pool** | 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 | **radius attribute nas-ip-address address** *ip_address* **nexthop-forwarding-address** *ip_address* **vlan** *vlan_id* | 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 | **[no] logical-port-statistics** | 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 | **vlan-map next-hop** *ipv4_address* | 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。 |

| CLI モード | コマンド | 説明 |
|---|---|---|
| イーサネットポートコンフィギュレーション モード | **vlan** *vlan_id* | 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 | **[no] shutdown** | 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。 |
| サブスクライバコンフィギュレーション モード | **ip vlan** *vlan_id* | 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 | **bind interface** *interface_name context_name* | 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 | **[no] ingress-mode** | 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 | **priority** *value* | 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 | **[no] shutdown** | 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 | **vlan-map interface** *if_name context_name* | 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 *25 : 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 | **clear port** *slot/port* **vlan** *vlan_id* | 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 | **show logical-port utilization table vlan { 5-minute \| hourly }** | 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 | **show port info** *slot/port* **vlan** *vlan_id* | 以前に設定された VLAN ID の NPU カウンタを表示します。 |



第 25 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

サービスプロバイダは、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-DI 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、RFC 4364（*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。

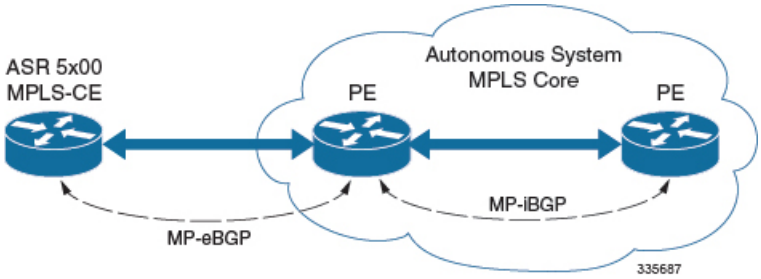
BGP/MPLS ソリューションでは、次のシナリオがサポートされています。

また、VPC-DI 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 （396 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-DI 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 *8 : VPC-DIMPLS-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。

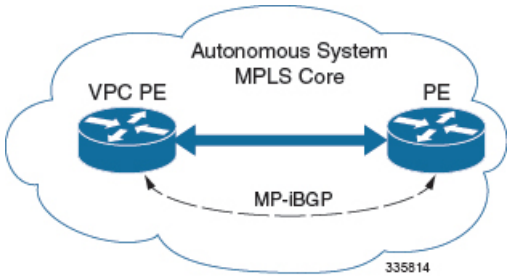
EBGP 接続を使用すると、PE は、IBGP ピアから学習したルート内のネクストホップ IP アドレスとラベルを変更してから、MPLS-CEにアドバタイズすることができます。この場合、MPLS-CE は、ルートをアドバタイズして学習するために、MP-eBGP だけを使用します。直接接続 EBGP ピアリングのため、Label Distribution Protocol（LDP）および Resource Reservation Protocol（RSVP）は必要ありません。このシナリオの MPLS-CE は、PE との間で（MP-eBGP 接続を介して学習された）単一のラベルをプッシュ/ポップします。

# VPC-DI PE として

## 概要

このシナリオでは、VPC-DI 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 *9 : VPC-DI PE* として




VPC-DI 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-DI は、次の 2 つのラベルを追加するように設定できます。

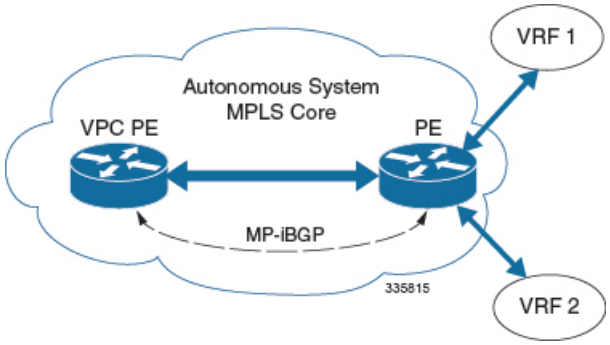
- LDP または RSVP TE から学習した外部ラベル（RSVP トラフィックエンジニアリング）
- MP-iBGP から学習した内部ラベル

このソリューションは、VPC-DI 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-DI 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-DI 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 *10 :* 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 node1_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-DI 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になります。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識別子（AFI）と後続のアドレスファミリ識別子（SAFI）の フィールドは、2 と 128 にれぞれ設定されます。

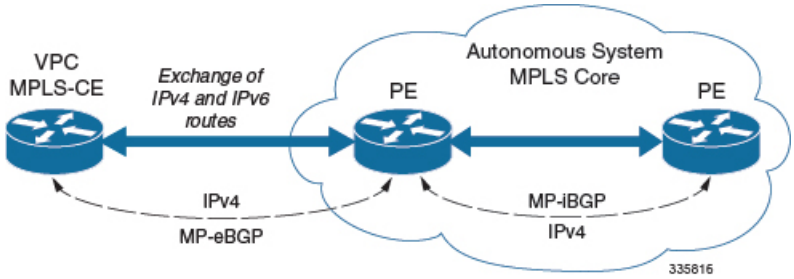
IPv6 VPN トラフィックは、IPv4 トンネリングを介して BGP スピーカーに転送されます。BGP スピーカーは、8 オクテットの RD がゼロに設定され、16 オクテットの IPv6 アドレスがアドバタイズルータの IPv4 アドレスを含む IPv4 マッピング IPv6 アドレス（RFC 4291）としてエンコードされている VPN-IPv6 アドレスを含むネクスト ホップ ネットワーク アドレス フィールドをピアにアドバタイズします。これは、VPNv6 ルートを交換するために EBGP ピアリングのみが使用されることを前提としています。

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。

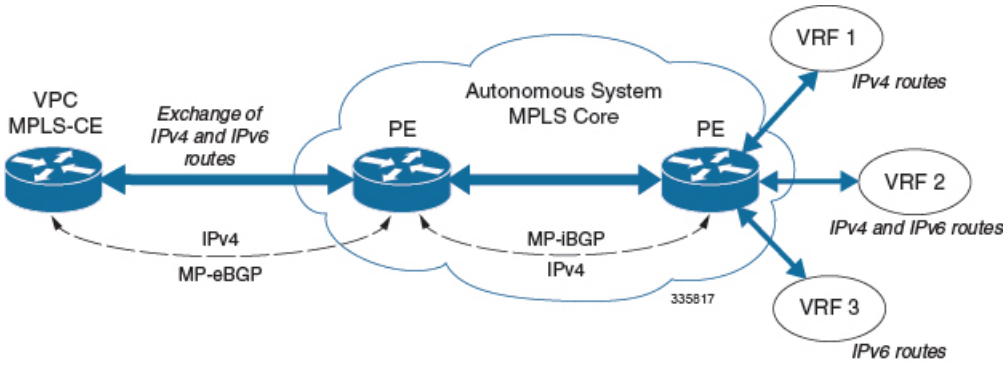
**図 *11 : VPNv6* に対する *IPv6-RD* サポート**




# 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 *12 : 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-noauthip context-name Gi_ce
  ipv6 address prefix-pool vrf3-v6pool
exit
aaa-group amazon-group
  radius ip vrf vrf2
aaa group default
exit
gtpp 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 *26 : 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（IPv4/IPv6）コンフィギュレーション モード | **neighbor** *ip_address* **activate** | 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（IPv4/IPv6）コンフィギュレーション モード | **neighbor** *ip_address* **send community { both \| extended \| standard }** | 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（IPv4/IPv6）コンフィギュレーション モード | **redistribute connected** | 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（VPNv4）コンフィギュレーション モード | **neighbor** *ip_address* **activate** | 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（VPNv4）コンフィギュレーション モード | **neighbor** *ip_address* **send community { both \| extended \| standard }** | 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 ルート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（VRF）コンフィギュレーション モード | **neighbor** *ip_address* **activate** | 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（VRF）コンフィギュレーション モード | **neighbor** *ip_address* **send community { both \| extended \| standard }** | 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 ルート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（VRF）コンフィギュレーション モード | **redistribute connected** | 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 | **address-family { ipv4 vrf** *vrf_name* **\| vpnv4 }** | 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 | **address-family { ipv6 vrf** *vrf_name* **\| vpnv6 }** | BGP で VPNv6 アドレスファミリと 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 | **ip vrf** *vrf_name* | 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。 |

| 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 | **route-distinguisher {** *as_value* **\|** *ip_address* **}** *rd_value* | 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。 |
| BGP IP VRF コンフィギュレーション モード | **route-target { both \| import \| export } {** *as_value* **\|** *ip_address* **}** *rt_value* | 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 | **ip pool** *pool_name addr_range* **vrf** *vrf_name* **[ mpls-label input** *inlabel1* **output** *outlabel1 outlabel2* **]** | 指定された VRF にプールを設定します。このパラメータは、ネクストホップパラメータで指定する必要があります。*inlabel1* は、このプールを宛ての着信トラフィックを識別する MPLS ラベルです。*outlabel1* および *outlabel2* は、このプールからサブスクライバに対して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 | **ip vrf** *vrf_name* | 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 | **ipv6 pool** *pool_name* **vrf** *vrf_name* | プールを VRF に関連付けます。<br>**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す。 |
| コンテキスト コンフィギュレーション モード | **mpls bgp forwarding** | 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 | **mpls exp** *value* | 3 ビット MPLS EXP ヘッダーのゼロ値を使用して、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 ( **mpls map-dscp-exp dscp** *n* **exp** *m* コマンドを使用)。<br><br>**mpls exp** 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 | **mpls ip** | 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 | **radius change-authorize-nas-ip ip_address** *ip_address* **{ encrypted \| key }** *value* **port** *port_num* **mpls input** *inlabel* **output** *outlabel1 outlabel2* | 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。 *inlabel* は着信 COA トラフィックを識別します。*outlabel1* および *outlabel2* は、COA 応答に追加する MPLS ラベルを指定します。*outlabel1* は内部出力ラベル、*outlabel2* 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 | **mpls ip** | このインターフェイスで IP パケットのダイナミック MPLS 転送を有効にします。 |
| Exec モード | **clear ip bgp peer** | BGP セッションをクリアします。 |

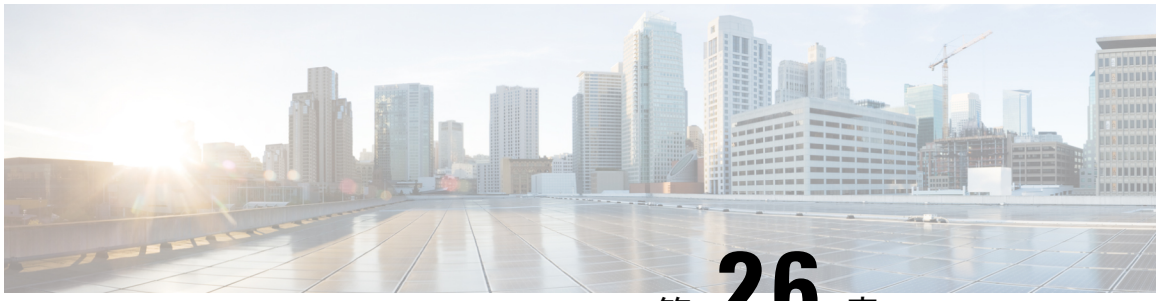
| CLI モード | コマンド | 説明 |
|---|---|---|
| Exec モード | **lsp-ping** *ip_prefix_FEC* | 指定された転送等価クラス（FEC）のMPLSラベルスイッチドパス（LSP）接続を確認します。その後にIPv4またはIPv6のFECプレフィックスが続く必要があります。 |
| Exec モード | **lsp-traceroute** *ip_prefix_FEC* | パケットが宛先に転送されるときに実際にたどるMPLS LSPルートを検出します。その後にIPv4またはIPv6のFECプレフィックスが続く必要があります。 |
| IP VRF コンテキスト コンフィギュレーション モード | **mpls map-dscp-to-exp dscp** *dscp_bit_value* **exp** *exp_bit_value* | 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。 |
| IP VRF コンテキスト コンフィギュレーション モード | **mpls map-exp-to-dscp exp** *exp_bit_value* **dscp** *dscp_bit_value* | 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 | **protocol ldp** | 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。 |

| 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 | **advertise-labels { explicit-null \| implicit-null }** | 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 | **discovery { hello { hello-interval** *seconds* **\| hold-interval** *seconds* **} \| transport-address** *ip_address* **}** | 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 | **enable** | 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 | **router-id** *ip_address* | LDP ルータ ID を設定します。 |
| MPLS-LDP コンフィギュレーション モード | **session timers { hold-interval** *seconds* **\| keepalive-interval** *seconds* **}** | LDP セッションパラメータを設定します。 |

表 *27 : 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 | **show ip bgp neighbors** | 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 | **show ip bgp vpnv4 { all \| route-distinguisher \| vrf }** | 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 | **show ip bgp vpnv6** | 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 | **show ip bgp vpnv6 { all \| route-distinguisher \| vrf }** | 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 | **show ip pool** | 設定された VRF を含むプールの詳細を表示します。 |

| CLI モード | コマンド | 説明 |
|---|---|---|
| Exec モードの show コマンド | **show mpls cross-connect** | 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 | **show mpls ftn [ vrf** *vrf_name* | 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 | **show mpls ftn [ vrf** *vrf_name* **]** | 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 | **show mpls ilm** | 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 | **show mpls ldp** | MPLS LDP 情報を表示します。 |
| Exec モードの show コマンド | **show mpls nexthop-label-forwarding-entry** | 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 26 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、次の内容で構成されています。



## 概要

Content Server Selection（CSS）は、モバイルサブスクライバによって提供されるか、モバイルサブスクライ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（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、サブスクライバ 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、サブスクライバプロファイル（または宛先コンテキストの APN プロファイル）を通じてサブスクライバに適用されます。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

すべてのサブスクライバの 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 （408 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：個々のサブスクライバへの ACL の適用（オプション） （409 ページ）の説明に従って、個々のサブスクライバに ACL を適用します。

**ステップ 3** オプション：複数のサブスクライバへの ACL の適用（オプション） （409 ページ）の説明に従って、複数のサブスクライバに単一の ACL を適用します。

**ステップ 4** オプション：複数のサブスクライバへの APN を介した ACL の適用 （342 ページ）の説明に従って、APN を介して複数のサブスクライバに 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、サブスクライ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、サブスクライ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ライ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。

# 個々のサブスクライバへの ACL の適用（オプション）

個々のサブスクライバへの 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「個々のサブスクライバへの *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クライバに適用されます。サブスク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クラ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クライバ。
- サブスク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クライバを「デフォルト」のプロファイルとして使用するように設定することで促進されたすべてのサブスクライバ。

## サブスクラ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ラ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「 サブスクラ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ライバへの ACL の適用（オプション）

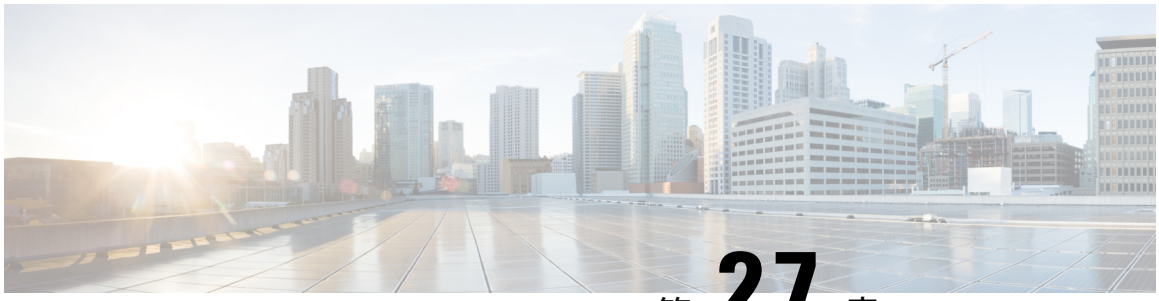
さまざまなシステムサービスによって「デフォルト」プロファイルとして使用されるように ACL をサブスクライバに適用する方法については、「アクセスコントロールリスト」の章の「サービス指定のデフォルトのサブスクライバへの *ACL* の適用」の項を参照してください。

# APN を介した複数のサブスクライバへの 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クライバに適用されます。サブスク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クライバパケットには、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クライバに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クライバへの単一 *ACL* の適用」の項を参照してください。



第 27 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、何の前触れもありません。

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

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）に必要になる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。アカウンティングおよび課金関連の情報のみがチェックポインティングされ、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）
- 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）
- 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ティングリカバリ
- 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

**重要** その他の可能なセッションリカバリおよびシャーシ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ライ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。
- サブスクライバデータの最小の統計情報セット。アカウンティング情報を確実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 （416 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&quest; [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 （416 ページ）

上級ユーザはテキストユーザ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ス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

               ----sessmgr----       ----aaamgr----          demux
 cpu state     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: 01ca11b1                  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
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:                              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。

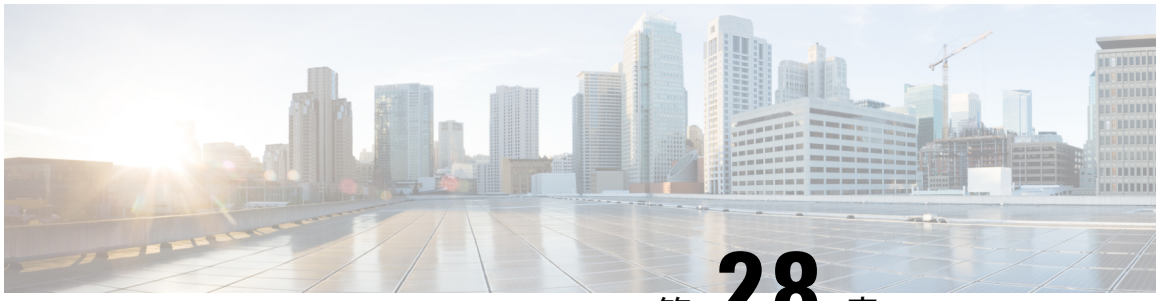
```
RCT stats details (Last 1 Actions)

 #  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 （419 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
```

```
    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 28 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-DI のコンテキストでは、シャーシは VPC-DI インスタンスです。ICSR は、同様に設定された VPC DI 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

ICSR 機能は、サブスクライ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-DIL2 と L3 の両方の ICSR をサポートします。

ICSR は冗長仮想シャーシを使用して実装されます。各 VPC-DI インスタンスの仮想シャーシは、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。両方の仮想シャーシが同じ AAA サーバ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、アクティブシャーシからスタンバイシャーシ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シがアウトオブサービスになると、スタンバイシャーシ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。

仮想シャーシは、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、プライマリシャーシとバックアップシャーシのアクティブな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（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

VPC DI 仮想シャーシは、定期的に Hello メッセージを使用して ICSR 通信をサポートするように設定されています。これらのメッセージは、各仮想シャーシのアクティブ CF によって送信され、現在の状態をピアに通知します。Hello メッセージには、設定や優先順位などの仮想シャーシに関する情報が含まれています。dead 間隔は、シャーシの CF ピアから受信される Hello メッセージの時間制限を設定するために使用されます。スタンバイシャーシの CF が dead 間隔の間にアクティブなシャーシ内のアクティブな CF から Hello メッセージを受信しない場合、スタンバイ仮想シャーシはアクティブ状態に移行します。

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

チェックポイントメッセージは、アクティブな仮想シャーシ内のアクティブ CF からスタンバイ仮想シャーシ内のアクティブ CF に送信されます。これらのメッセージは特定の間隔で送信され、そのシャーシがアクティブになった場合にスタンバイシャーシ上のセッションを再作成するために必要なすべての情報が含まれます。セッションがチェックポイント期間を超えると、そのセッションでチェックポイントデータが収集されます。

詳細については、付録の「*ICSR* のチェックポインティング」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 *28 : srp CLI* コマンド**

| コマンド | 説明 |
|---|---|
| **srp disable nack micro-chkpt-cmd** | 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 メッセージの送信を無効にします。完全なチェックポイントを送信すると、SRP の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。 |
| **srp initiate-audit manual-with-sync** | 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。 |
| **srp initiate-switchover** | 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。 |
| **srp reset-auth-probe-fail** | 認証プローブモニタの障害情報を 0 にリセットします。 |
| **srp reset-diameter-fail** | Diameter モニタの障害情報を 0 にリセットします。 |
| **srp terminate-post-process** | スイッチオーバー後の処理を強制的に終了します。 |
| **srp validate-configuration** | アクティブシャーシの設定を検証します。 |
| **srp validate-switchover** | 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** リリース 20.0 以降の場合、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 *29 : show srp* コマンド**

| コマンド | 説明 |
|---|---|
| **show srp audit-statistics** | 外部監査の統計情報を表示します。 |
| **show srp call-loss statistics** | 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。 |
| **show srp checkpoint statistics** | 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。 |
| **show srp info** | サービス冗長プロトコル情報（コンテキスト、シャーシの状態、ピア、接続状態など）を表示します。 |
| **show srp monitor** | SRP モニタ情報を表示します。 |
| **show srp statistics** | SRP 統計情報（送信された hello メッセージ、設定の検証、リソースメッセージ、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

AAA サーバは、認証プローブメカニズムを使用してモニタされます。認証プローブが有効な応答を受信した場合、AAA サーバは稼働していると見なされます。AAA サーバの設定で指定された **max-retries count** に達した場合、AAA サーバは停止していると見なされます。設定済

みの AAA サーバ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、アクティブシャーシ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シが設定済みの AAA サーバのモニタに失敗した場合、次のいずれかが発生するまでアクティブシャーシのままになります。

- 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、シャーシ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、設定されたシャーシ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シ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、アクティブシャーシとスタンバイシャーシで **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 2 つの VPC-DI インスタンスまたは UGP 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。

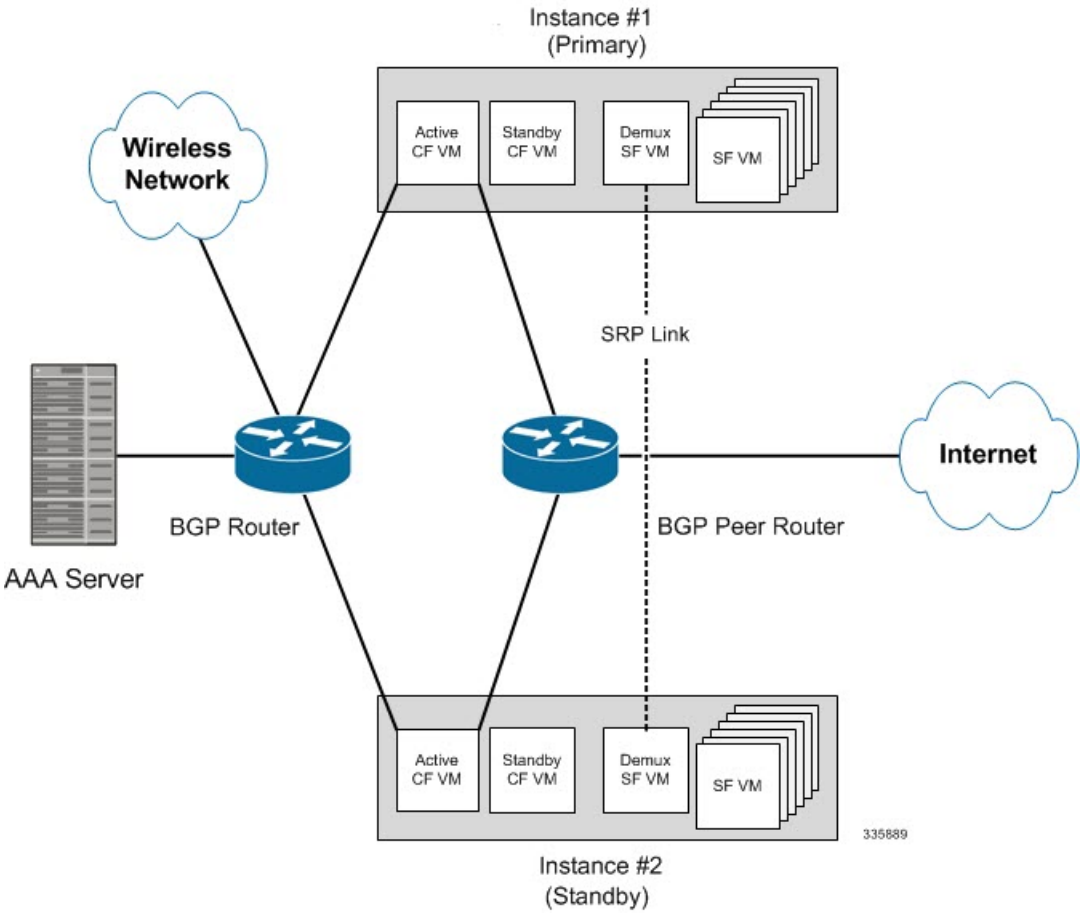
- 3 つのコンテキスト：
  - **冗長性**：プライマリシャーシとバックアップシャーシの冗長性を設定します。
  - **送信元**：指定された nas-ip-address の AAA 設定は、HA にバインドされたインターフェイスの IP アドレスであるか、または同じコンテキスト内で設定されたコアネットワークサービスである必要があります。
  - **接続先**：PDN へのモニタリングとルーティングを設定します。
- ボーダー ゲートウェイ プロトコル（BGP）：ICSR はルート修飾子を使用してシャーシの優先順位を決定します。

**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、各シャーシ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、両方のシャーシにログインし、**show license information** コマンドを実行します。「シャーシ間セッションリカバリ」を探します。シャーシ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できます。

次の図は、ICSR のネットワークを示しています。

図 *13 :* 仮想化された *StarOS* の *ICSR* のネットワーク

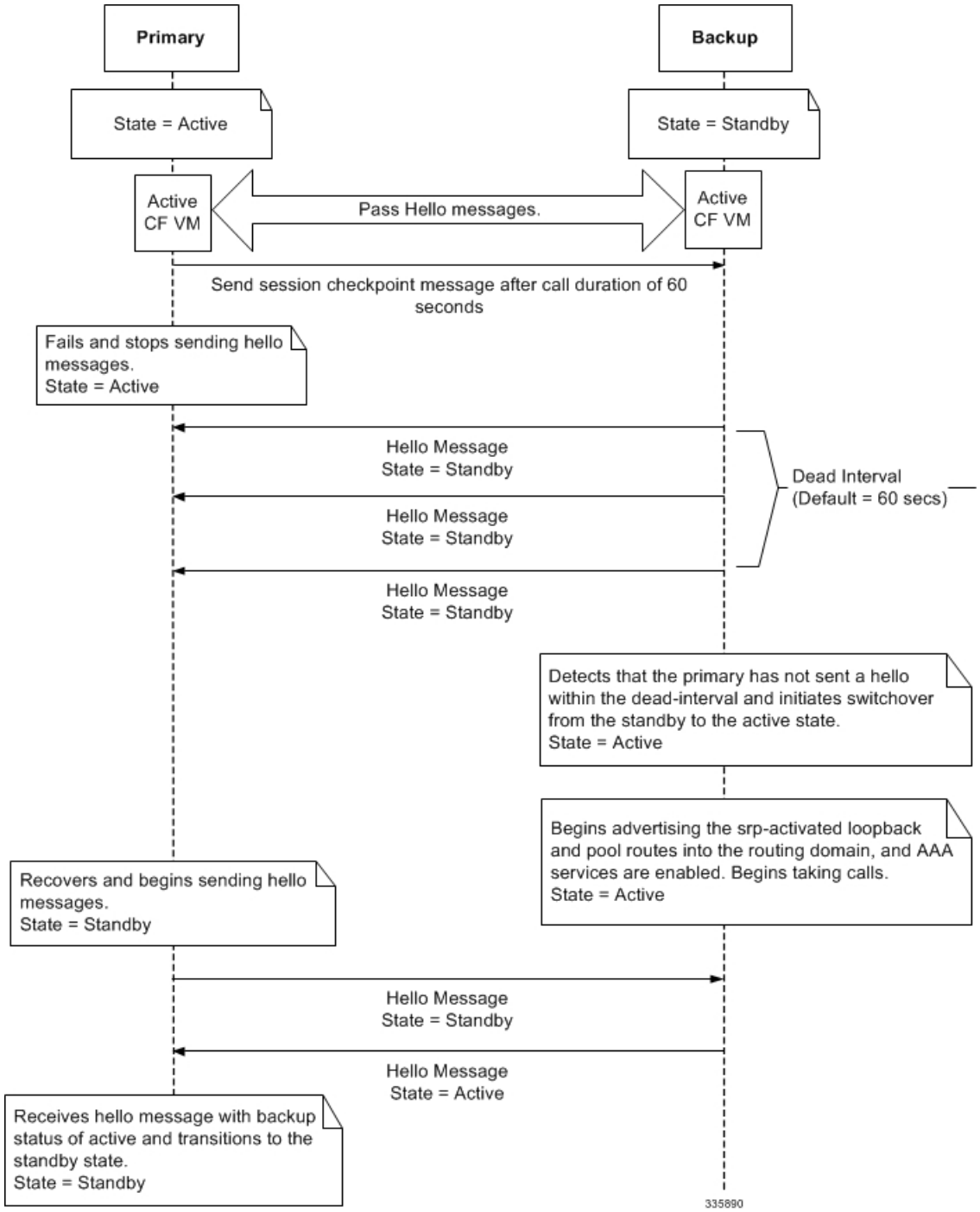



# ICSR の運用

このセクションには、ICSR の運用フローが表示されます。

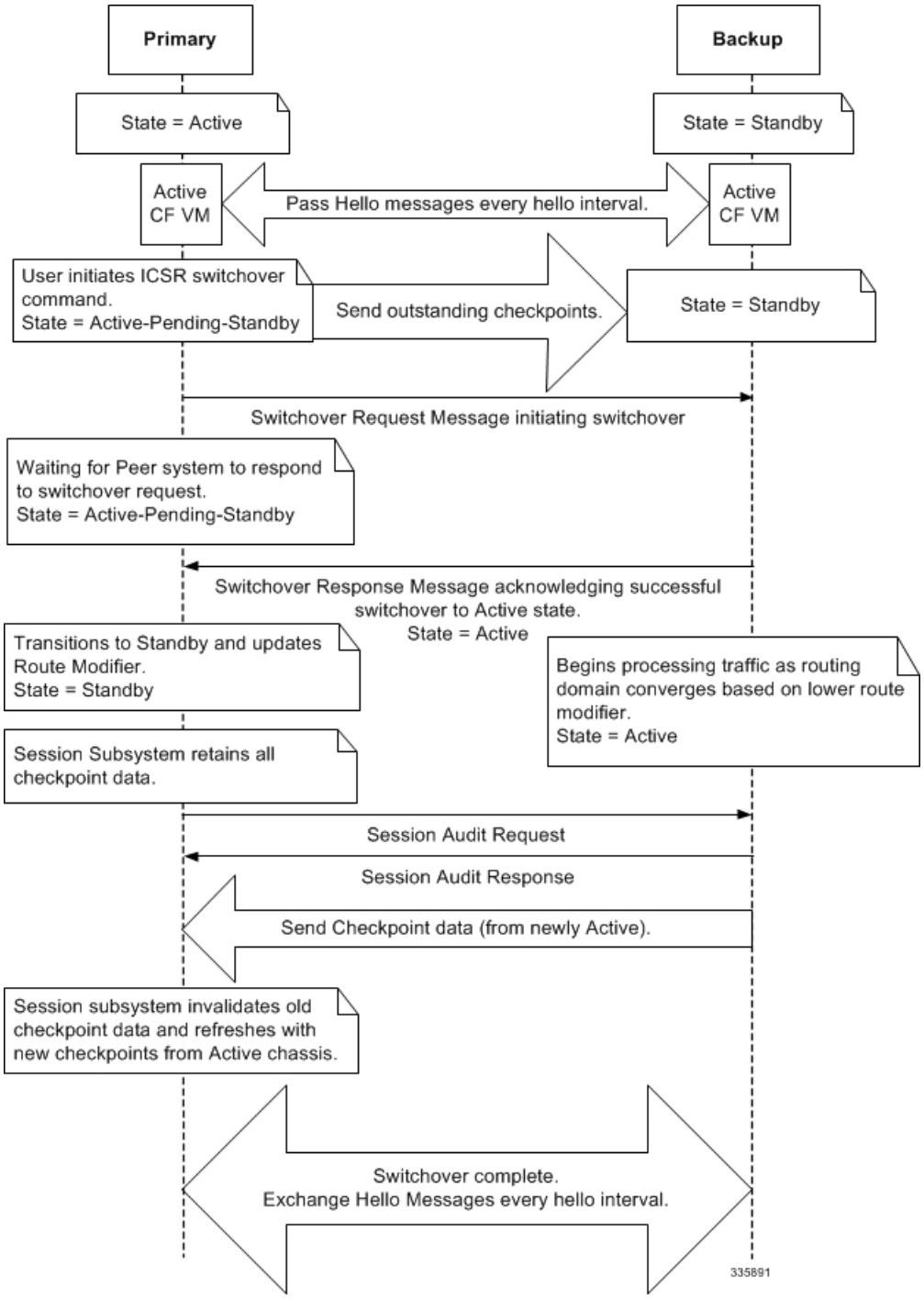
次の図は、プライマリ障害による ICSR プロセスフローを示しています。

図 *14 : 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 *15 : ICSR* プロセスフロー（手動スイッチオーバー）



# シャーシの初期化

StarOS が各 VPC DI 仮想シャーシで同時に初期化されると、アクティブ CF は Hello メッセージを互いに送信します。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 CF の 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、CF 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 つのセットがベンダーコード）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、CF MAC アドレス *00-02-43-03-1C-2B* を持つ VPC-DI 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

1 つの仮想シャーシがアクティブ状態で、もう 1 つがスタンバイ状態である場合、両方の仮想シャーシがアクティブ CF を介して Hello メッセージを各 hello 間隔で送信します。チェックポイントのセッション期間を超えるサブスクライバセッションは、スタンバイシャーシに送信されるチェックポイントメッセージに含まれます。チェックポイントメッセージにはサブスクライバセッション情報が含まれているため、アクティブなシャーシがアウトオブサービスになると、バックアップシャーシがアクティブになり、サブスクライバセッションの処理を続行できるようになります。スタンバイシャーシでサブスクライバセッション情報が更新されるたびに、追加のチェックポイントメッセージがさまざまな間隔で発生します。

SRP コンフィギュレーションモードの**checkpoint session** コマンドには、次のことを可能にする多数のキーワードが含まれています。

- SRP ペイロードメッセージに使用する圧縮アルゴリズムのタイプを設定します。
- 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、シャーシが待機する時間を設定します。チェックポイントは、IMS や非 IMS セッション用に個別に設定できます。
- 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（完全なチェックポイント）の送信間隔を設定します。

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

## シャーシスイッチオーバー

アクティブ仮想シャーシがサービスを提供しなくなった場合、スタンバイシャーシがHelloメッセージの送信を継続します。スタンバイシャーシが dead 間隔内にアクティブ CF から Hello メッセージへの応答を受信しない場合、スタンバイシャーシがスイッチオーバーを開始します。スイッチオーバー中に、スタンバイシャーシ内のアクティブ CF は SRP によってアクティブ化されたループバックとルーティングドメインへのプールルートのアドバタイズを開始します。シャーシがアクティブになると、既存の AAA サービスと、チェックポイント情報を備えていたサブスクライバセッションの処理は続行されますが、新しいサブスクライバセッションを確立することもできます。

プライマリ仮想シャーシが復帰すると、設定されているピア内のアクティブ CF に Hello メッセージを送信します。ピアは応答を送信し、シャーシ内のアクティブ CF 間の通信を確立し、設定情報を含む Hello メッセージを送信します。プライマリシャーシはバックアップシャーシの状態がアクティブであることを示す Hello メッセージを受信した後、スタンバイに移行します。Hello メッセージの送信は各ピアに対して継続され、チェックポイント情報がアクティブなシャーシからスタンバイシャーシに定期的に送信されるようになります。

シャーシのスイッチオーバーが発生すると、セッションタイマーが回復します。セッションの潜在的な損失の可能性と、更新が過渡的なチェックポイント更新プロセス時に損失する可能性を回避するため、アクセスゲートウェイのセッションリカバリはフルライフタイムで再作成されます。

# 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。

**重要** *StarOS* の *17.0* よりも前のリリースでは、L2TP コールをサポートするシャーシに ICSR を設定しないでください。

次に説明する手順は、下記のも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 （433 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 （444 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 （446 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 （447 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 （447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。

# 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 （434 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 （434 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 （441 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 （444 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 VPC-DI インスタンス内のアクティブ CF の IP アドレスにバインドします。

**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 VPC-DI インスタンス内のアクティブ CF にバインドしてください。バックアップ VPC-DI 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シにシャーシ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、シャーシ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、両方のシャーシが同じシャーシモードで誤って設定されている場合に、どのシャーシがアクティブになるかが決定されます（シャーシの初期化（431 ページ）を参照）。優先順位の高いシャーシの方が小さい番号になります。各シャーシ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シの IP シャーシを **peer-ip-address** として、プライマリシャーシに入力します。プライマリシャーシの IP アドレスを **peer-ip-address** として、バックアップシャーシに割り当てます。
- **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 ガートタイマー、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**：クラスセレクタ 1 PHB
- **cs2**：クラスセレクタ 2 PHB
- **cs3**：クラスセレクタ 3 PHB
- **cs4**：クラスセレクタ 4 PHB
- **cs5**：クラスセレクタ 5 PHB
- **cs6**：クラスセレクタ 6 PHB
- **cs7**：クラスセレクタ 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。

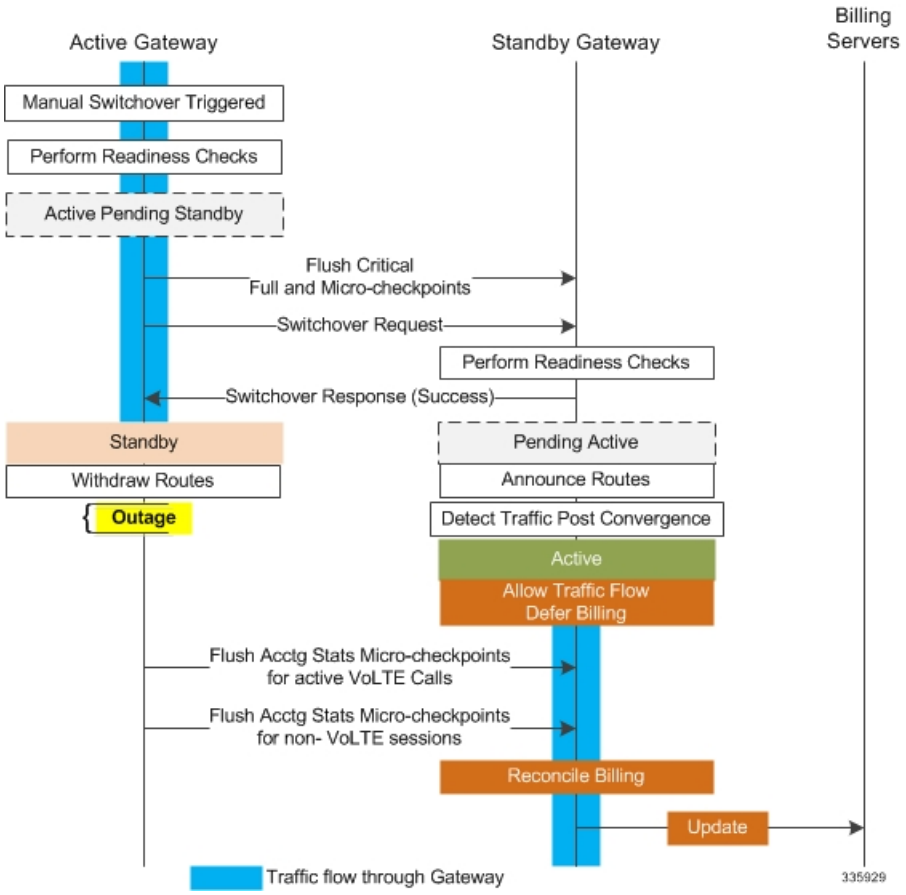
### ICSR スイッチオーバー時の非 VoLTE トラフィックの許可

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、アカウンティ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。

- スイッチ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ティング情報をすべて受信すると、外部課金サーバ 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 *16 :* 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 コールのアカウンティング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**（**アカウンティングの維持**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ライ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、アクティブシャーシ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、アカウンティ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シ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。
- **アカウンティング フラッシュ**：このフラッシュの段階で、シャーシ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ティ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、アカウンティ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 スイッチ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確実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）の受信に応答して 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 （444 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 （445 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します。

**ステップ 3** BGP 設定の確認 （445 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 （446 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 （446 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します。

**ステップ 3** サブスクライバをデフォルトモードに設定 （446 ページ）の手順に従って、サブスクライバモードを **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 （447 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
```

## サブスクライバをデフォルトモードに設定

サブスクライバのモードを**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
```

## サブスクライ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** またはシャーシ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。ICSR 展開をアップグレードまたはダウングレードする前に、Cisco TAC 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 *17 : ICSR* ソフトウェアアップグレード：パート *1*

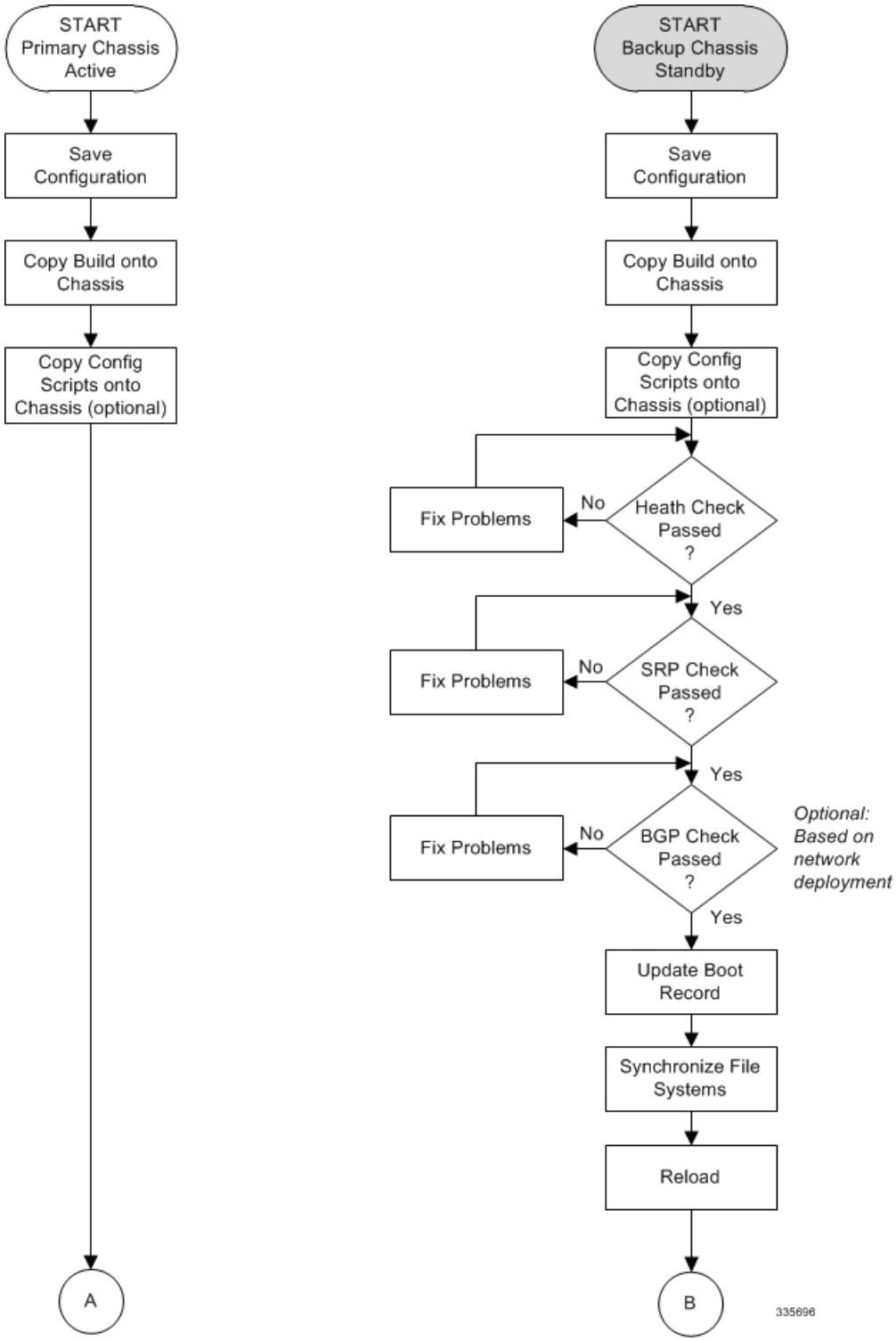


図 *18 : ICSR* ソフトウェアアップグレード：パート*2*

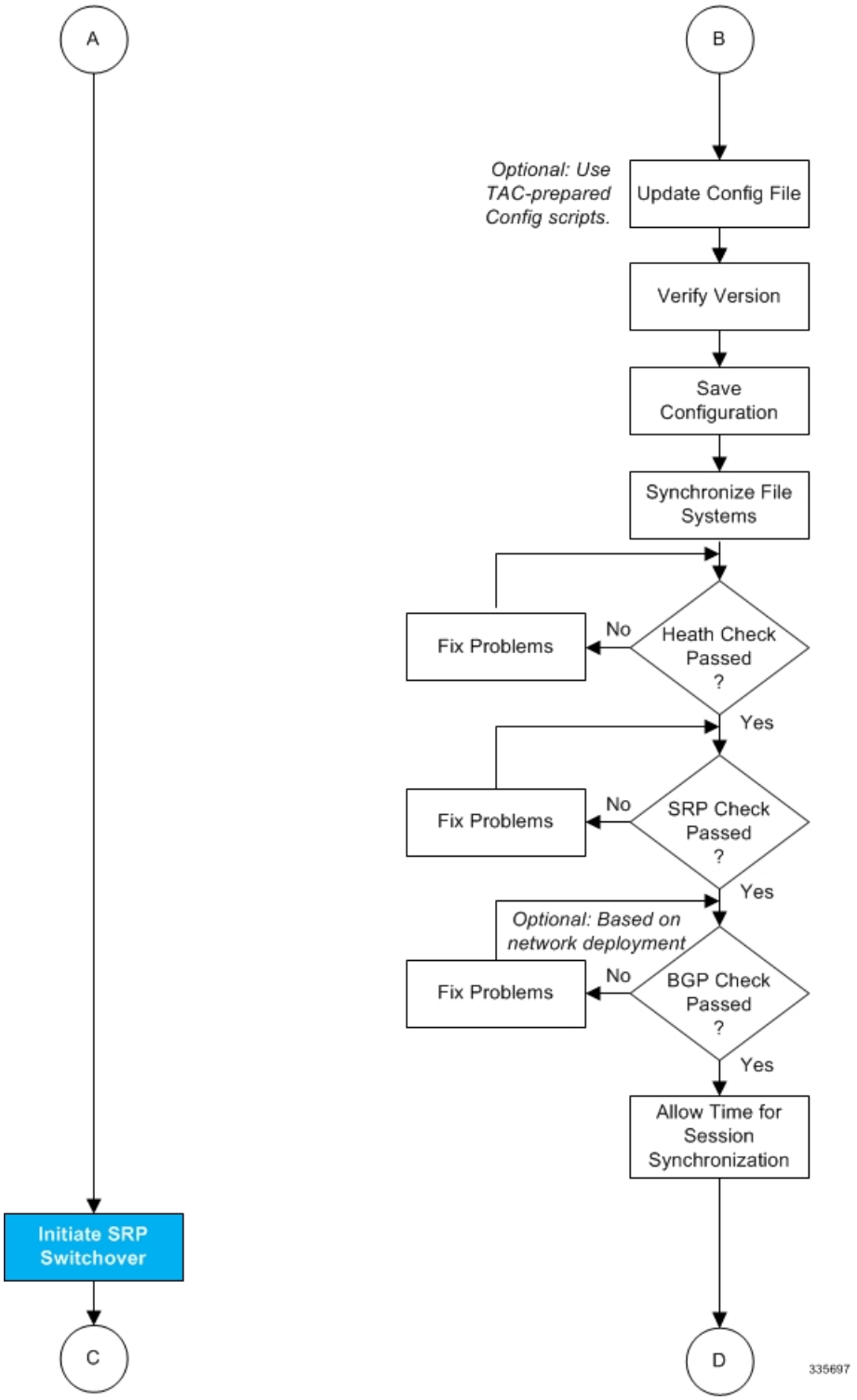


図 *19 : ICSR* ソフトウェアアップグレード：パート*3*

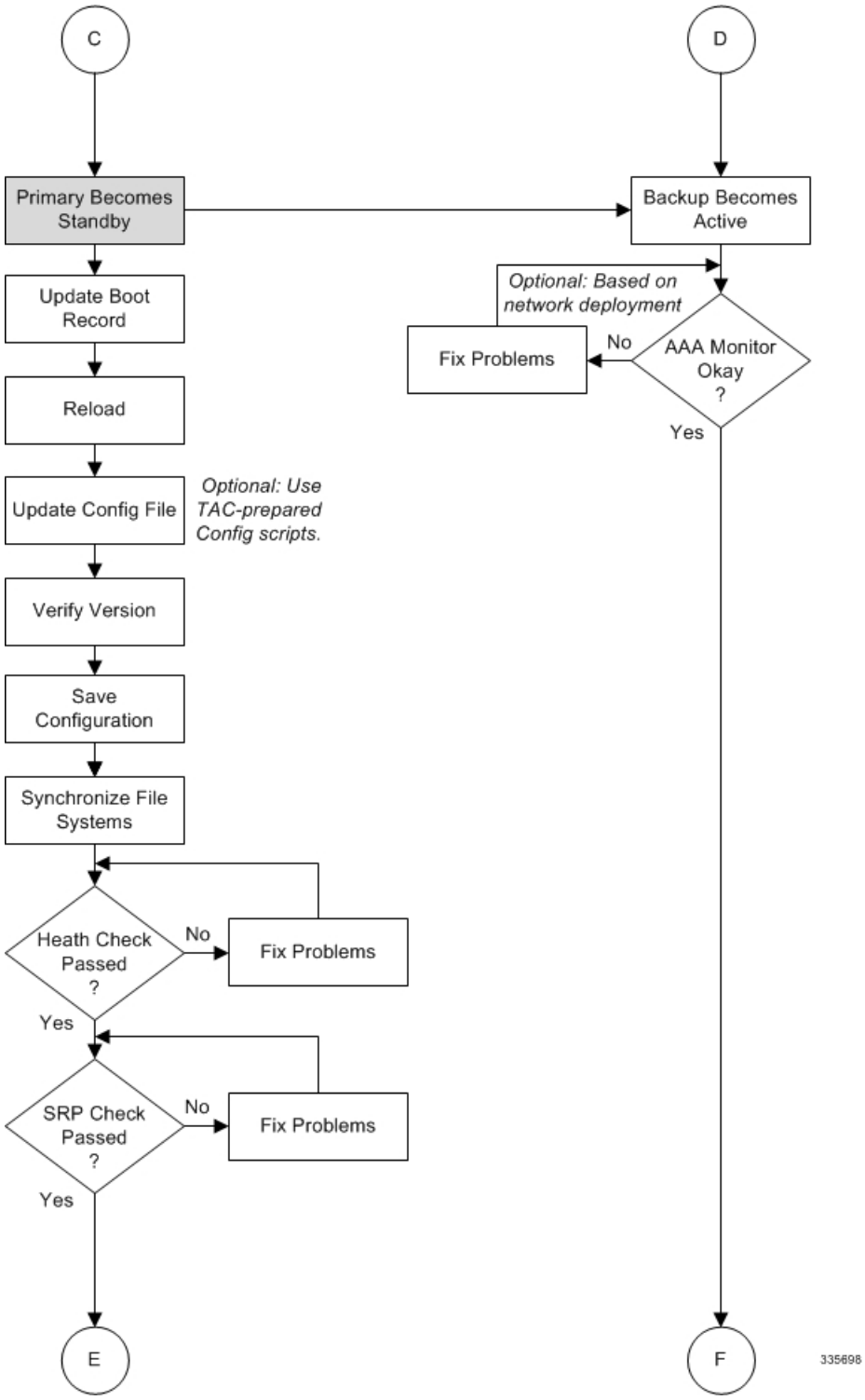
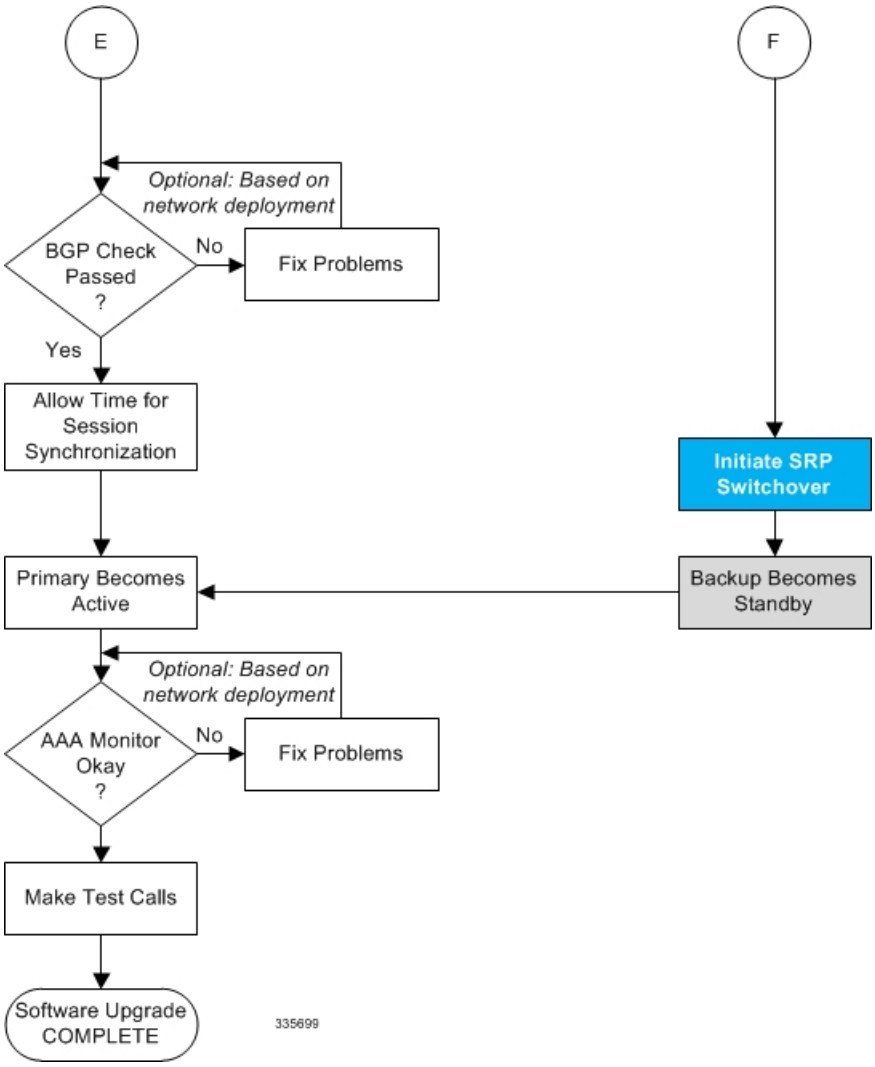


図 *20 : ICSR* ソフトウェアアップグレード：パート *4*




# 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、シャーシに再転送します。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
```

**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 （457 ページ）
- 正常性チェックの実行 （455 ページ）
- SRP チェックの実施 （456 ページ）
- BGP チェックの実施 （456 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クライバの数と、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 モニタステータスの確認

ネットワーク展開で AAA サーバ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 モニタチェックを実行します。システムが AAA サーバ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 サーバと通信できない場合、詳細については「AAA モニタ」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 （456 ページ）
- StarOS の再起動 （457 ページ）
- 設定ファイルの更新 （457 ページ）
- ソフトウェアバージョンの確認 （457 ページ）
- 設定ファイルの保存 （457 ページ）
- ファイルシステムの同期 （457 ページ）
- 正常性チェックの実行 （455 ページ）
- SRP チェックの実施 （456 ページ）
- BGP チェックの実施 （456 ページ）
- セッションの同期の待機 （458 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、サブスク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 （455 ページ）

第 29 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。



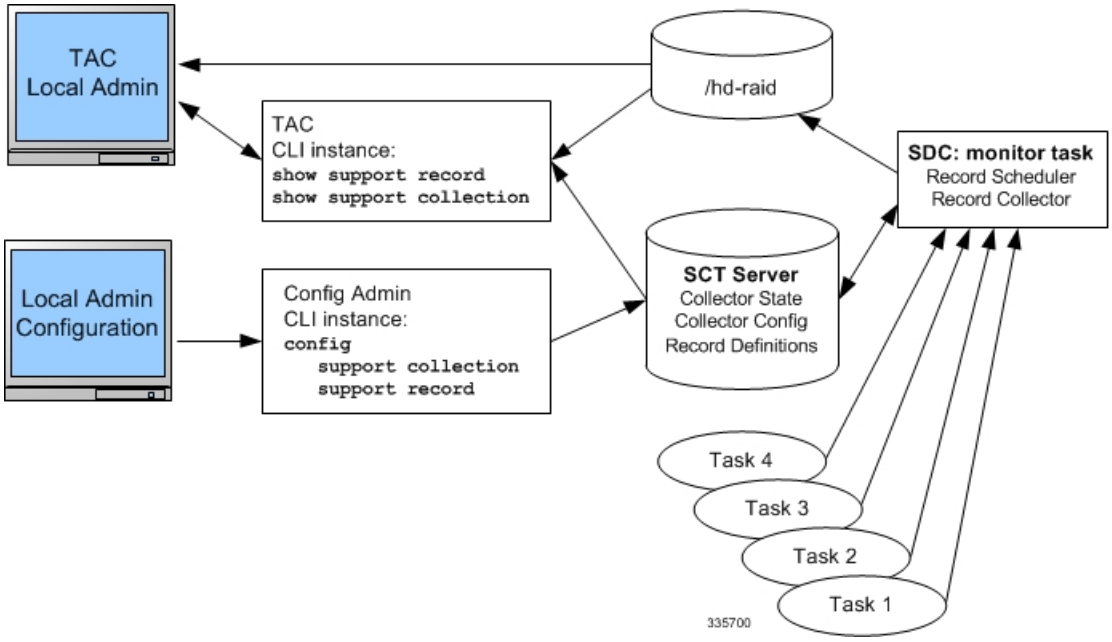
## 概要

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実行します。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もあります。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、メッセンジャの要求を表し、データの主要なフローを示します。

図 *21 : SDC* のタスクとプロセス <




# 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） （466 ページ）を参照してください。

# SDR 収集設定の表示

ユーザが [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

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。

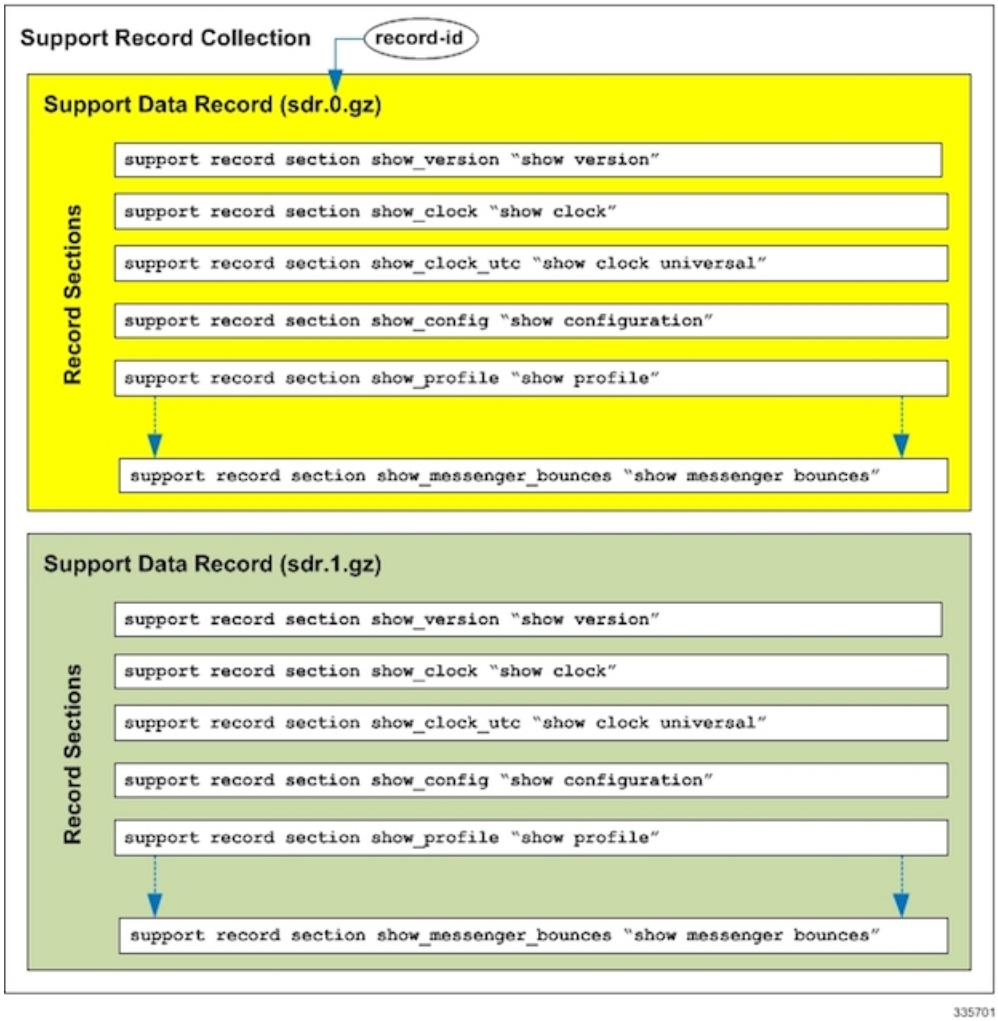
SDC の周期は、グローバル コンフィギュレーション モードで **support collection schedule** コマンドによって設定されます。SDR が保存されると、SDC は、**support collection** コマンドを介して指定されたスリープ時間間隔を待ってから、別の SDR を収集します。

**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 *22 :* 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 （467 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 ]**

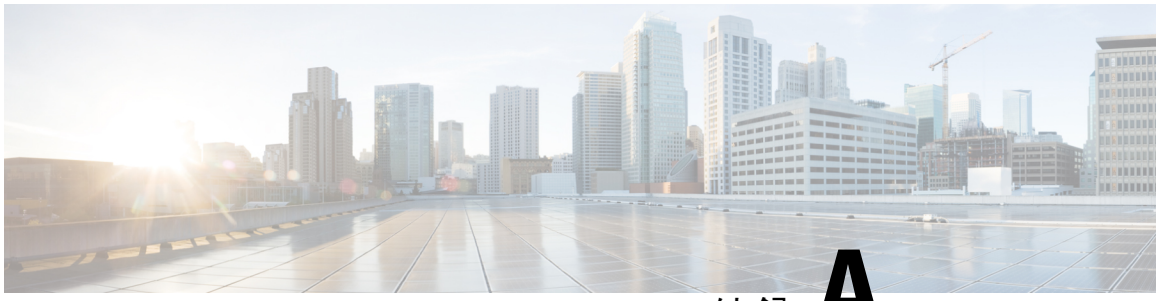
**show support collection** コマンドは、SDC アクティビティに関する情報を表示します。このコマンドは、最後にスケジュールされた収集の開始時刻、最後にスケジュールされた収集の期間、収集がまだ進行中かどうかなどの情報を表示します。さらに、現在保存されている一連の SDR レコード ID、それぞれのタイムスタンプ、各 SDR のサイズを一覧表示します。

```
[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

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の 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-DI インターフェイスとポートのルール

この項では、サービス機能（SF）VM のコンテキストトラフィックに使用される vNIC イーサネットポートに関するルールについて説明します。

## vNIC イーサネットポート

- すべての割り当てられた論理インターフェイスに一意の名前を付けて、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識別します。異なるコンテキストの論理インターフェイスに、同じ名前を付けることができます。
- 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、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。
- すべての論理インターフェイスに、有効な IP アドレスとサブネットを割り当てます。
  - コンテキスト内の各論理インターフェイスに一意の IP アドレスを指定します。異なるコンテキストの論理インターフェイスは、同じ IP アドレスを持つことができます。
  - 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。
  - 共有サブネットは、プライマリとセカンダリの両方の IP アドレスに対して設定できます。たとえば、次の設定が有効です。

```
interface intf1
   ip address 190.20.20.2 255.255.255.0
   ipv6 address 2001::34/64 secondary
   exit
interface intf2
   ip address 190.20.20.3 255.255.255.0
   ipv6 address 2001::35/64 secondary
   exit
interface intf3
   ipv6 address 2001::36/64
   ip address 190.20.20.4 255.255.255.0 secondary
   exit
```

- 論理インターフェイスは 1 つのコンテキストでのみ設定できますが、1 つのコンテキストで複数のインターフェイス（最大 512）を設定できます。
- 1 つの論理インターフェイスに最大 256 のアクセスコントロールリスト（ACL）ルールを適用できます。
- すべてのポートが *slot*/*port* によって識別されます。VPC-DI の場合、*slot* は仮想シャーシ内の CF または SF VM に対応 します（VPC-DIインスタンス）。ハイパーバイザは VPC-DI インスタンスの初期設定時に、各 VM に一意のスロット番号を割り当てます。スロット番

号 1 と 2 は CF VM に割り当てられ、スロット番号 3 ～ 32 は SF VM に割り当てられます。CF はポート 1 のみをサポートします。

各 SF は、1 番 ～ 12 番の 12 個の vNIC をサポートし、対応する仮想イーサネットポート 10 番 ～ 21 番を使用します。SF ポート番号 10 を設定する必要があります。

- サブスクライバトラフィック用の各 vNIC ポートには、最大 1,024 の VLAN タグを含めることができます。
- 論理インターフェイスは、*<card/slot/port>* によって識別される単一の物理ポートで単一の VLAN を使用するように制限されています。

# 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
  - リリース *15.0* よりも前：最大 16 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。
  - リリース *15.0* 以降：デマルチプレクサ 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。
  - リリース *15.0* よりも前：最大 32 の IPv6 プールを 1 つのコンテキスト内に設定できます。
  - リリース *15.0* 以降：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 プールの実際の数は、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、プールの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。

- 同時サブスクライ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（シャーシあたり 48,000）。
  - コンテキストあたり 6,000 のプールルート（シャーシあたり 6,000）

- *18.5* よりも前のリリース：コンテキストあたり 5,000 のプールの明示的ホストルート（シャーシあたり 6,000）
- リリース *18.5* 以降：コンテキストあたり 24,000 のプールの明示的ホストルート（シャーシあたり 24,000）
- コンテキストあたり 64 のルートマップ

• BGP

- リリース *12* および *14*：コンテキストあたり 16,000 の BGP プレフィックスを学習/アドバタイズできます（シャーシあたり 64,000）
- リリース *15* および *16*：コンテキストあたり 32,000 の BGP プレフィックスを学習/アドバタイズできます（シャーシあたり 64,000）
- リリース *17*、*18* 以降：コンテキストあたり 64,000 の BGP プレフィックスを学習/アドバタイズできます（シャーシあたり 64,000）
- コンテキストあたり 64 の EBGP ピアの設定が可能（シャーシあたり 512）
- コンテキストあたり 16 の IBGP ピア
- シャーシ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

• OSPF

- シャーシあたり 200 の OSPF ネイバー
- コンテキストあたり 10,000 の OSPF ルート（シャーシあたり 64,000）

• MPLS

リリース *19.x* ～リリース *21.6*

- 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
  - リリース *15.0* よりも前：コンテキストあたり 250 の Virtual Route Forwarding（VRF）テーブル（シャーシあたり 1,024 または 2,048（リリース 14.0+）の VRF）
  - リリース *15.0* 以降：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）
  - リリース *15.0* よりも前：シャーシあたり 256,000 のプレフィックス/フレームルートと、MR（モバイルルータ）あたり最大 8 つの動的に学習されたプレフィックス
  - リリース*15.0* 以降：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 サーバグループの場合はコンテキストあたり 128 の AAA サーバサーバは、アカウンティング、認証、課金サーバ、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 サーバグループを設定できます。
  - AAA サーバグループあたり 128 のサーバ（アカウンティング、認証、課金サーバ、またはその組み合わせ）
  - AAA サーバグループモードあたり 1,600 のサーバ（アカウンティング、認証、課金サーバ、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（サーバグループあたりプライマリ 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す。

- 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、ASN GW、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。

- *17.0* よりも前のリリースでは、StarOS は最大 512 グループをサポートしています。
- リリース *17.0* 以降では、StarOS は最大 2048 グループをサポートしています。

（注）

- *max_num* は、1 ～ 10 の整数（*18.2* よりも前のリリース）

  リリース *18.2x*
  - QVPC-DI: 32
  - QVPC-SI: 32

  リリース *21.4x*
  - QVPC-DI: 64
  - QVPC-SI: 64
  - ASR 5500: 24

- 設定の確認と保存 （149 ページ）の章の説明に従って、設定を保存します。

- リリース *20.0* 以降では、StarOS は最大 64000 グループをサポートしています。

# VPC-DI ネットワークの依存関係

この項では、VPC-DI インスタンスの StarOS ゲートウェイの機能を使用するために必要なサービスとネットワークデバイスの概要を示します。

## ルータ

VPC-DI ネットワークをサポートしているルータは、次をサポートする必要があります。

- IPv4 および IPv6 インターフェイス（注：各ルーティングコンテキスト内では、StarOS は各 IP インターフェイスを一意のサブネットに配置する必要があります）。
- スタティック ルート
- オプション：OSPFv2 および/または OSPFv3（IPv4/IPv6 のマルチキャストサポートが必要）
- eBGP（外部ボーダー ゲートウェイ プロトコル）
  - Multihop
  - Multipath
  - IPv4 アドレスファミリ
  - IPv6 アドレスファミリ
  - オプション：企業スタイル設定のための VPNv4 と VPNv6
  - これらのルータは eGBP を介して IP プール、ルート、サービス インターフェイス アドレスなどを学習します。
- ルートをフィルタ処理するためのルーティングポリシー。
- MPLS のカプセル化と転送（注：企業スタイル設定にのみ必要）
- ICMP と ICMPv6 のサポート
- BFD（双方向フォワーディング検出）（失敗を迅速に検出するため）
- ECMP（等コストマルチパス）

## 外部ネットワークの依存関係

- RADIUS または Diameter サービス
- DNS サービス
- NTPv4 サービス
- SNMP サービス
- syslog サービス

- SSH/SFTP サービス（バルク統計情報および管理用）
- 任意：TACACS+ サービス
- 課金サービス（CDR/EDR）
- VLAN カプセル化
- ICSR の要件：
  - プライマリとバックアップの仮想シャーシ間の専用接続
  - eBGP
  - 2 つの仮想シャーシ間のレイヤ 2 L2 スイッチ機能（シャーシの状態が変更された場合に、GARP/MAC アドレスの変更を処理するため）

# 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<br>シャーシあたり 64 | コンテキストあたり 16<br>シャーシあたり 64 | コンテキストあたり 16<br>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<br>シャーシあたり 3,200 万 | コンテキストあたり 1,600 万<br>シャーシあたり 3,200 万 | コンテキストあたり 1,600 万<br>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<br>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<br>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<br>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<br>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<br>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<br>シャーシあたり 5,000（IPv4 と IPv6 の組み合わせ） |

| 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 +） |
|---|---|---|---|
| BGPプレフィックスの数 | コンテキストあたり 16,000<br>シャーシあたり 64,000 | コンテキストあたり 32,000<br>シャーシあたり 64,000 | コンテキストあたり 64,000<br>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）<br>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）<br>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。 |
| 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<br>シャーシあたり 512 | コンテキストあたり 64<br>シャーシあたり 512 | コンテキストあたり 64<br>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<br>シャーシあたり 48,000 | コンテキストあたり 8,000 ラベル<br>シャーシあたり 48,000 | コンテキストあたり 8,000 ラベル<br>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<br>シャーシあたり 48,000 | コンテキストあたり 8,000 ラベル<br>シャーシあたり 48,000 | コンテキストあたり 8,000 ラベル<br>シャーシあたり 48,000 |

| 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 +） |
|---|---|---|---|
| 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<br>コンテキストあたり 32 IPv6 IP トンネルインターフェイス<br>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<br>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<br>コンテキストあたり 32 IPv6 IP トンネルインターフェイス<br>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<br>コンテキストあたり 256 ループバック インターフェイス<br>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<br>コンテキストあたり 32 IPv6 IP トンネルインターフェイス<br>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<br>コンテキストあたり 256 ループバック インターフェイス<br>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 |
| 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<br>シャーシあたり 64,000 | コンテキストあたり最大 10,000<br>シャーシあたり 64,000 | コンテキストあたり最大 10,000<br>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）<br>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） |

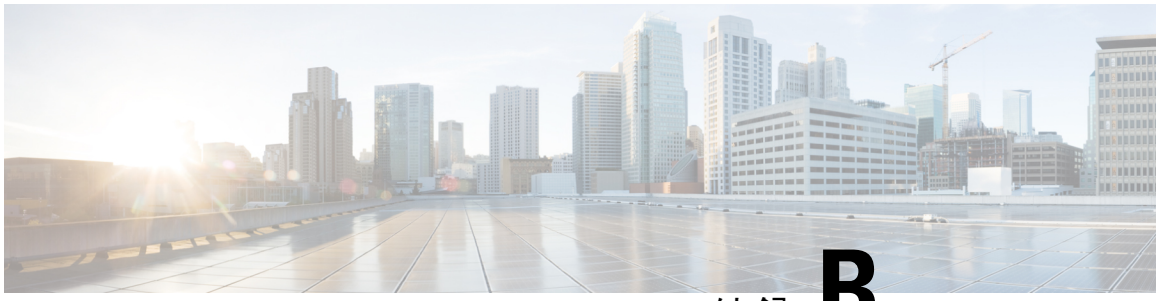
| パラメータ | スケーリング番号（リリース 12.x、14.x） | スケーリング番号（リリース 15.x、16.x） | スケーリング番号（リリース 17.x、18.x、19.x、20.x +） |
|---|---|---|---|
| プールルートの数 | コンテキストあたり 6,000（シャーシあたり 6,000） | コンテキストあたり 6,000（シャーシあたり 6,000） | コンテキストあたり 6,000（シャーシあたり 6,000） |
| ルートの数（フレームルートを除く） | コンテキストあたり 64,000 | コンテキストあたり 64,000 | コンテキストあたり 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 |
| VRF の数 | コンテキストあたり 250<br>シャーシあたり 2,048<br>APN の制限は 1024/シャーシであり、VRF の制限と一致しません。 | コンテキストあたり 300<br>シャーシあたり 2,048<br>（注）<br>• VRF の制限と APN の制限は同一であると想定しています。<br>• 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。 | コンテキストあたり 300<br>シャーシあたり 2,048<br>注：VRF の制限と APN の制限は同一であると想定しています。<br>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。 |

| 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 |
| 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<br>シャーシあたり 9600万 | コンテキストあたり 3200 万<br>シャーシあたり 9600万 | コンテキストあたり 3200 万<br>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。

エンジニアリングルール



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

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

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）[1]：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

[1] 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 （488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

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、モニタします。（ASR 5500 のみ） |

| タスク | 説明 | 機能 |
|---|---|---|
| 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。 |

| タスク | 説明 | 機能 |
|---|---|---|
| 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（**router bgp** 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。 |

| タスク | 説明 | 機能 |
|---|---|---|
| ospf | Open Shortest Path First | OSPF ルーティングプロトコル（**router ospf** 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。 |
| ospfv3 | Open Shortest Path First | OSPFv3 ルーティングプロトコル（**router ospfv3** 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。 |

| タスク | 説明 | 機能 |
|---|---|---|
| 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（**router rip** 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）<br>（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<br>（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。 |

| タスク | 説明 | 機能 |
|---|---|---|
| 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。 |
| npusim | NPU シミュレータ<br>（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ライ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。 |
| | | **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。<br>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。 |

| タスク | 説明 | 機能 |
|---|---|---|
| a11mgr | A11 マネージャ | 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。 |
| aaamgr | 認証、許可、およびアカウンティング（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ライバと管理ユーザに対して、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。 |

| タスク | 説明 | 機能 |
|---|---|---|
| aaaproxy | 認証、許可、およびアカウンティング（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 **gtpp single source** コマンドが設定されるたびに起動します。<br>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ティングサーバ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 **demux card** 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。 |
| 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 **active-charging service** 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。 |

| タスク | 説明 | 機能 |
|---|---|---|
| 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 とベアラーコンテキストを維持します。 |
| | | MBMS ベアラーセッションの BMSC サーバへの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、MBMSUEとMBMSベアラーコンテキスト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、シャーシ全体に対して 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。 |

| タスク | 説明 | 機能 |
|---|---|---|
| egtpemgrr | 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。 |
| 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 sgw-ingress または pgw-ingress 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。 |
| | | 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。 |

| タスク | 説明 | 機能 |
|---|---|---|
| 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。 |
| gtpumgr | GPRS トンネリングプロトコルユーザ（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。 |

| タスク | 説明 | 機能 |
|---|---|---|
| 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。 |
| 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 はシャーシに 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。 |

| タスク | 説明 | 機能 |
|---|---|---|
| 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。 |
| | | 新しいサブスクラ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。<br>- セッションコントローラ<br>- MME マネージャ<br>- 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。 |

| タスク | 説明 | 機能 |
|---|---|---|
| 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、サブスクラ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ラ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。 |
| | | サーバモードでは、RADIUS サーバとして機能し、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がサポートをサポートします。 |
| | | 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。 |

| タスク | 説明 | 機能 |
|---|---|---|
| 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。 |
| 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。 |

| タスク | 説明 | 機能 |
|---|---|---|
| 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。 |
| 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。 |

| タスク | 説明 | 機能 |
|---|---|---|
| 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。シャーシには、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。 |
| | | 個々のサブスクライ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ライバ情報を維持します。 |
| | | サブスクライバごとに 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。 |

| タスク | 説明 | 機能 |
|---|---|---|
| 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 も保存されます。 |
| | | 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 proclets 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。 |

| タスク | 説明 | 機能 |
|---|---|---|
| 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担当します。afioは、ローカルafmgrと他のローカルafioインスタンスとの非メッセンジャのプロセス間通信（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。 |
| cspctrl | 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、モバイルサブスクライバによって提供されるか、モバイルサブスクライ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。<br>コンテンツサーバ（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サーバは、コンテキスト コンフィギュレーション モードの **css server** コマンドを介して設定されます。 |
| | | CSSコントローラはCSSマネージャを作成しません。CSS マネージャは、VPNマネージャによって停止および開始されます。CSSマネージャは、コンテキストごとに自動的に作成されます。 |
| 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 サーバへのキープアライブを管理します。 |
| | | サブスクライバのCSS関連情報を取得します。 |
| | | CSSサーバがダウンすると、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間でデータを再配布したりします。 |

| タスク | 説明 | 機能 |
|---|---|---|
| dcardctrl | ドーターカードコントローラ<br>[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<br>[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。 |
| 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。 |

| タスク | 説明 | 機能 |
|---|---|---|
| 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 は適切なサーバプログラムを起動して接続を処理します。<br>**注**：リリース 20.0 以降の信頼できる StarOS ビルドでは、FTP および 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。 |
| 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。 |

| タスク | 説明 | 機能 |
|---|---|---|
| lagmgr | リンク集約グループマネージャ<br>[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 | メッセンジャデーモン | 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 | メッセンジャ 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 | メッセンジャプロセスの一環として、メッセン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 **ntp server** CLI コマンドを使用して 1 つ以上の NTP サーバが設定されている場合に有効になります。 |

| タスク | 説明 | 機能 |
|---|---|---|
| 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 プレーンをモニタします。 |
| 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 **server sshd** 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<br>**注**：リリース 20.0 以降の信頼できる StarOS ビルドでは、FTP および Telnet はサポートされていません。 |

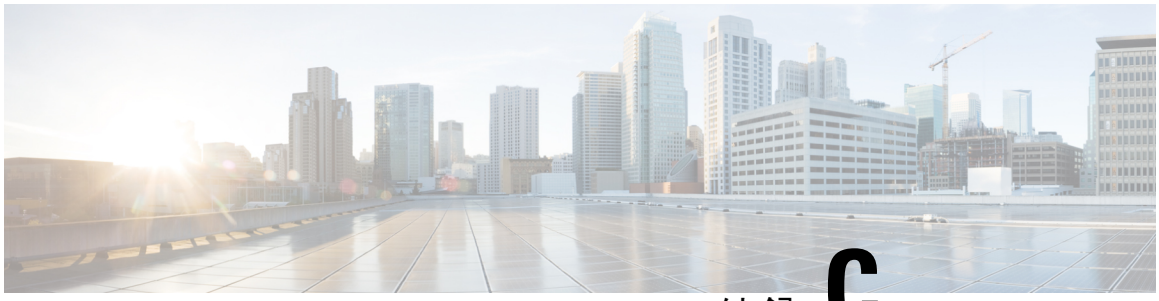
# 管理プロセス

**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を処理します。 |

| タスク | 説明 | 機能 |
|---|---|---|
| orbs | ORBEM サービス<br>[ASR 5500 のみ] | ORBS タスクは、ORB 要素マネージャ（ORBEM）とも呼ばれています。<br>要素管理システム（EMS）は、セキュアな IIOP 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。<br>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 サーバに渡されます。 |
| orbns | ORBEM 通知サービス<br>[ASR 5500 のみ] | イベントの発生を EMS サーバに通知します。 |
| | | このような EMS サーバを登録し、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 サーバに通知します。 |
| 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 | 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。 |

StarOS のタスク



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。



## 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 エンジン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。ConfD Manager は、新しいユーザを認証/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。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の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、CLIアクセス（外部からNETCONF）へのすべての変更をYANGモデルでサポートされる設定オブジェクトに手動で適用する必要があります。

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コマンドのサブセットを組み込みます。

このリリースでは、YANGモデルはNSOを介して限られた一連のECSコンフィギュレーションコマンドをサポートしています。詳細については、NSOユーザマニュアルおよびサポートされている StarOS ECS 設定コマンド （534 ページ）を参照してください。

# 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)# ?
  autosave-config      - Automatically saves current configuration to the URL specified
                         whenever a change is applied through ConfD interfaces
  confd-user           - Configures the default login user with full administrator rights
                         for the ConfD server.
  end                  - Exits configuration mode and returns to Exec Mode
  exit                 - Exits current configuration mode, returning to previous mode
  no                   - Enables/Disables the followed option
```

**autosave-config** キーワードおよび **confd-user** キーワードはオプションです。

NETCONF プロトコルを無効にするには、グローバル コンフィギュレーション モードの **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。

## autosave-config

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、CLI ベースのモデルの ConfD インターフェイスを介して、変更が NSO によって適用されるたびに、現在の ConfD 設定を指定された URL に自動的に保存します。デフォルトでは、このコマンドは無効になっています。

コマンドシンタックスは、**autosave-config** <*url*>、**[ file: ]{/flash | /usb1 | /hd-raid | /sftp} [** /<*directory*>**] /**<*filename*> です。

## 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。このユーザ名は、CLI ロギング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。

# 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 のユーザマニュアルを参照してください。

# 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
      confd-user NETCONF
      autosave-config /flash/config.cfg
    #exit
    subscriber default
    exit
    aaa group default
    #exit
    gtpp group default
    #exit
  #exit
end
```

注：

- **confd**：**ユーザ**と**自動保存設定**は任意です。**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 } | subscriptions }**

注：

**show confdmgr** の出力例については、以下を参照してください。

- **confd** キーワードを指定すると、次のオプションが表示されます。
  - **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。
- **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。

```
[local]<host_name># show confdmgr

State Information
-----------------
State                Started
Subscriptions        2
Last successful id   1461-704882-705350
Last failed id       None
Autosave url         Not configured
Username             Not configured

Statistics
----------
Triggers                  1
Notifications             2
Successful notifications  2
Failed notifications      0
Unexpected                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 ConfD に CDB の内容をダンプして confdmgr に戻し、その結果、SCT による設定の同期が行われた回数（共有設定タスク）。
- [Notifications]：ConfD が confdmgr に更新を送信した回数。
- [Successful Notifications]：ConfD から受信した更新が正常に処理された回数。
- [Failed Notifications]：ConfD から受信した更新が正常に処理されなかった回数。成功した回数と失敗した回数は、常に通知の合計数と同じである必要があります。
- [Unexpected]：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 cbd** 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 cdb
active-charging service acs
exit
context local
 server confd
  autosave-config /tmp/ut_confdmgr_config.txt
  confd-user      confd_user
 exit
exit
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_confd
  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** 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
```

NETCONF および ConfD

```
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
NAME         CREATOR  CREATED                    CONTEXT
-----------------------------------------------------------
/rollback0   admin    2016-04-26T17:08:02-00:00  noaaa
/rollback1   admin    2016-04-26T17:07:57-00:00  noaaa
/rollback2   admin    2016-04-26T17:07:49-00:00  noaaa
/rollback3   admin    2016-04-26T17:07:49-00:00  noaaa
/rollback4   admin    2016-04-26T17:07:47-00:00  noaaa
 |
 |
 V
/cli-history/admin.hist
/cli-history/root.hist
/global.data

[local]<host_name>#
```

**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 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                        6     staros
/context                                7     staros
[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。このリリースでは、「/active-charging」と「/context」の 2 つのサブスクリプションがあります。

# 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、YANG モデルでサポートされているコンフィギュレーション 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 rd1
      tcp any-match = TRUE
    #exit
    rulebase default
    #exit
  #exit
end
[local]<host_name>#
```

# clear confdmgr statistics

このコマンドは、**show confdmgr** コマンドの出力の「Statistics」セクションに一覧表示されているすべての内容をクリアします。これには次が含まれます。

- Triggers
- 通知（Notifications）
- Successful notifications
- Failed notifications

• Unexpected

# 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。

# 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
[localhost_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。

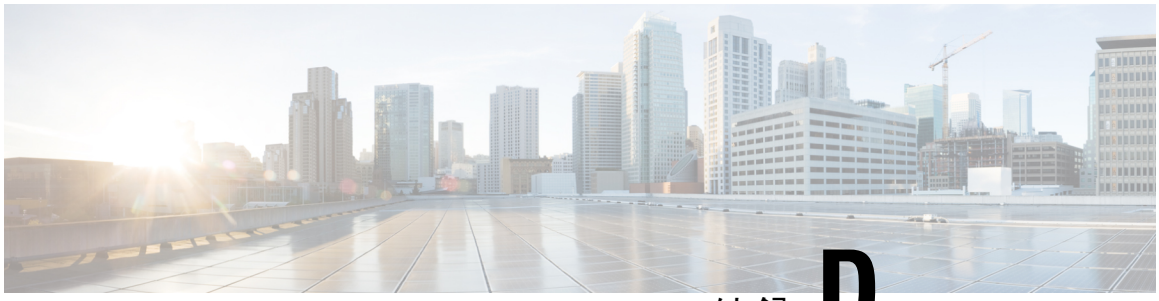
- ruledef <*ruledef_name*>
  - ip server-ip-address = *
  - tcp-ether-port = *
  - udp ether-port = *
  - tcp either-port-range = *
  - udp ether-port range = *

- 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 1
  - 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.

# ConfD のアップグレードサポート



付録 D

# ICSR のチェックポイント

この付録では、シャーシ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 シャーシ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

シャーシ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 がピアシャーシ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シからスタンバイシャーシに最初に送信され、その後は定期的に送信されます。スタンバイシャーシは、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（コール回線ポインタ）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、アクティブシャーシからスタンバイシャーシ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

## 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 チェックポインティング中に発生します。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 チェックポインティング中に発生します。
- **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、ベアラーの 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。

- ベアラーの作成または削除
- 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、他のサブスクライバと NAT IP アドレスを共有しているサブスクライバに対して、ポートチャン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。ポートチャンクは、そのサブスクラ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ポートチャン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 サブスクライバから解除されたときに送信されます。

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ライ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ライバに割り当てられたか、またはその割当をサブスクライ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 フローがサブスクライ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 データはサブスクライ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 フローがサブスクライバに対して作成または削除されたときに送信されます（データはサブスクライ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、サブスクライバに対して NAT が有効になっているが、（ルールの一致に基づいて）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**：Sb6 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 からアカウンティング要求（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 からアカウンティング要求（INTERIM-UPDATE）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**：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。
- **アカウンティング**：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_EOGRE_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。
- **アカウンティング**：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

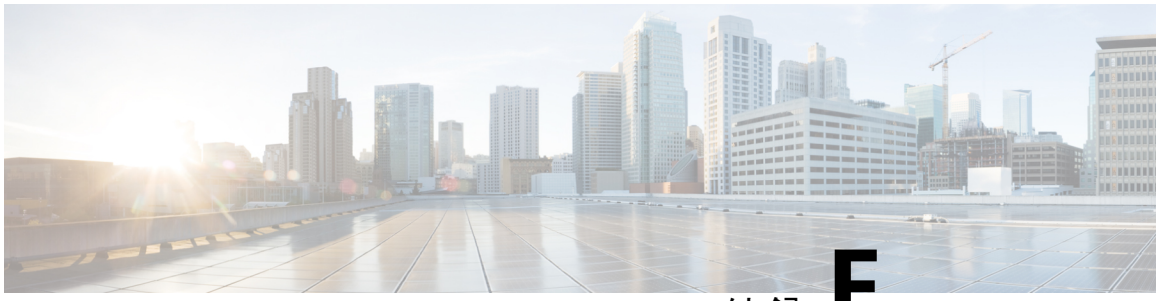
このマイクロチェックポイントは、サブスクライバの再認証が完了したときに、SaMOG セッション用に送信されます。

- **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 サブスクライ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、サブスクライ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**：サブスクライバを再許可するために、AAA サーバから 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-DI 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-DI 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" |

VPC-DI 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 |
|---|---|---|
| 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" |

VPC-DI 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 |
|---|---|---|
| 86 | 無効 | "show sgsn-service all" |
| 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 pdsnclosedrp-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" |

| 番号 | デフォルト SDR | コマンド文字列 |
|---|---|---|
| 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" |
| 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" |

VPC-DI 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 |
|---|---|---|
| 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" |
| 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" |

VPC-DI 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 |
|---|---|---|
| 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 henbgw-access-service statistics" |
| 167 | 無効 | "show henbgw-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" |
| 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" |

VPC-DI 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 |
|---|---|---|
| 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 a11mgr 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" |
| 202 | 有効 | "show demuxmgr statistics sgtpcmgr all" |
| 203 | 無効 | "show demuxmgr statistics imsimgr all" |

VPC-DI 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 |
|---|---|---|
| 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 gtpp accounting servers" |
| 212 | 無効 | "show gtpp statistics verbose" |
| 213 | 無効 | "show gtpp counters all" |
| 214 | 無効 | "show gtpp storage-server" |
| 215 | 無効 | "show gtpp storage-server statistics verbose" |
| 216 | 無効 | "show gtpp storage-server local file statistics verbose"<br>**重要** 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。 |

VPC-DI 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 |
|---|---|---|
| 217 | 無効 | "show gtpp storage-server local file counters all"<br>**重要** 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。 |
| 218 | 無効 | "show gtpp storage-server streaming file statistics verbose" |
| 219 | 無効 | "show gtpp storage-server streaming file counters all" |
| 220 | 無効 | "show gtpp group all" |
| 221 | 有効 | "show hd-storage-policy statistics all verbose" |
| 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 sntp statistics verbose" |
| 235 | 無効 | "show llc statistics verbose" |

| 番号 | デフォルト SDR | コマンド文字列 |
|---|---|---|
| 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" |
| 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" |

VPC-DI 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 |
|---|---|---|
| 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-charging edr-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" |

VPC-DI 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 |
|---|---|---|
| 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" |
| 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" |

VPC-DI 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 |
|---|---|---|
| 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" |
| 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" |

| 番号 | デフォルト SDR | コマンド文字列 |
|---|---|---|
| 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 srdbmgr all verbose" |
| 323 | 無効 | "show content-filtering category statistics" |
| 324 | 無効 | "show content-filtering category statistics facility srdbmgr 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" |

VPC-DI 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 |
|---|---|---|
| 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" |
| 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" |

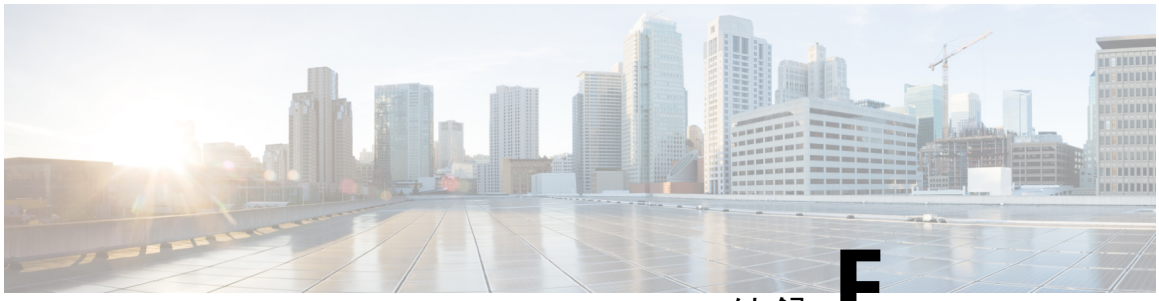
VPC-DI SDR CLI コマンドストリング

| 番号 | デフォルト SDR | コマンド文字列 |
|---|---|---|
| 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

VPC-DI SDR CLI コマンドストリング



付録 F

# VPC コマンド

この章では、StarOS リリース19.1 以降に VPC で導入または変更されたコマンドの詳細について説明します。

# access-type

このコマンドは、TAI データベースごとの NB-IoT RAT の設定に使用されます。

**Command_Product**

MME

**Command_Privilege**

管理者

**コマンド モード**

Exec > グローバル コンフィギュレーション > LTE ポリシー コンフィギュレーション > LTE TAI 管理データベース コンフィギュレーション

**configure > lte-policy > tai-mgmt-db** *db_name*

上記のコマンドシーケンスを入力すると、次のプロンプトが表示されます。

```
[local]host_name(tai-mgmt-db)#
```

**構文の説明**

**[ no ] access-type nb-iot**

**no**

TAI データベースに設定されているアクセスタイプを削除します。

**nb-iot**

TAI データベースのアクセスタイプを NB-IoT として設定します。

**使用上のガイドライン**

LTE TAI 管理データベース コンフィギュレーション モードは、システム上の LTE トラッキングエリア識別子（TAI）管理データベースを作成して管理するために使用されます。既存または新たに定義されたデータベースの TAI 管理データベース コンフィギュレーション モードを開始します。また、このコマンドは既存のデータベースを削除するためにも使用されます。このコマンドを使用して、TAC または TAC のグループのアクセスタイプを NB-IoT RAT として

設定します。3GPP 標準に従って、同じ TAC を EUTRAN と NB-IoT RAT の両方に帰属させることはできません。このコマンドはデフォルトでは無効になっています。デフォルトの RAT は WB-EUTRAN です。

### 例

次のコマンドのコマンドを使用して、アクセスタイプを NB-IoT に設定します。

```
access-type nb-iot
```

# access-type

このコマンドは、TAI オブジェクトごとに NB-IoT RAT を設定するために使用されます。

**Command_Product**

MME

**Command_Privilege**

管理者

**コマンド モード**

Exec > グローバル コンフィギュレーション > LTE ポリシー設定 > LTE TAI 管理データベース コンフィギュレーション > LTE TAI 管理オブジェクト設定

**configure > lte-policy > tai-mgmt-db** *db_name* **> tai-mgmt-obj** *obj_name*

上記のコマンドシーケンスを入力すると、次のプロンプトが表示されます。

`[local]host_name(tai-mgmt-obj)#`

**構文の説明**

**[ no ] access-type nb-iot**

**no**

TAI オブジェクトに設定されているアクセスタイプを削除します。

**nb-iot**

TAI オブジェクトのアクセスタイプを NB-IoT として設定します。

**使用上のガイドライン**

LTE TAI 管理オブジェクト コンフィギュレーション モードは、TAI データベースの LTE トラッキングエリア 識別子を作成および管理するために使用されます。このモードは、既存の LTE トラッキングエリア識別子（TAI）オブジェクト設定を作成、削除、または変更するために使用されます。このコマンドを使用して、TAC または TAC のグループのアクセスタイプを NB-IoT RAT として設定します。3GPP 標準に従って、同じ TAC を EUTRAN と NB-IoT RAT の両方に帰属させることはできません。このコマンドはデフォルトでは無効になっています。デフォルトの RAT は WB-EUTRAN です。

### 例

次のコマンドのコマンドを使用して、アクセスタイプを NB-IoT に設定します。

**access-type nb-iot**

# bfd

Bidirectional Forwarding Detection（BFD）のインターフェイスパラメータの設定。

**Command_Product**

すべて

**Command_Privilege**

セキュリティ管理者、管理者

**コマンド モード**

[Exec] > [Global Configuration] > [Context Configuration[ > [Ethernet Interface Configuration]

**configure > context** *context_name* **> interface** *interface_name* **broadcast**

上記のコマンドシーケンスを入力すると、次のプロンプトが表示されます。

```
[context_name]host_name(config-if-eth)#
```

**構文の説明**

```
[no] bfd { echo [echo-interval interval_num] | interval interval_num }
   min_rx milliseconds multiplier value
```

**no**

このインターフェイスで指定されたオプションを無効にします。

**echo**

BFD エコー モードをイネーブルにします。

BFD エコー モードは非同期 BFD で動作します。エコー パケットはフォワーディング エンジンによって送信され、検出を実行するために、同じパスで転送されます。反対側の BFD セッションはエコー パケットの実際のフォワーディングに関与しません。エコー機能およびフォワーディング エンジンが検出プロセスを処理するため、2 つの BFD ネイバー間で送信される BFD 制御パケットの数が減少します。

遅延エンジンがリモートシステムを介さずにリモート（ネイバー）システムの転送パスをテストするため、パケット間の遅延のばらつきが改善される可能性があり、それによってBFDバージョン 0 を BFD セッションの BFD 制御パケットを使用する場合に、障害検出時間を短縮できます。

**echo-interval *interval_num***

BFD エコーパケット間の送信間隔を指定します。デフォルトの間隔は 150 ms です。範囲は 0 ～ 999 ms です。（VPC のみ）

**interval *interval_num***

BFD パケット間の送信間隔（ミリ秒単位）を指定します。

- *17.0* より前のリリースでは、*interval_num*は 50 ～ 999 の整数です。（デフォルトは 50）
- *17.0* 以降のリリースでは、*interval_num*は 50 ～ 10000 の整数です。（デフォルトは 50）

**min_rx *milliseconds***

制御パケットの受信間隔をミリ秒単位で指定します。

- *17.0* より前のリリースでは、*milliseconds*は 50 ～ 999 の整数です。（デフォルトは 50）
- *17.0* 以降のリリースでは、*milliseconds*は 50 ～ 10000 の整数です。（デフォルトは 50）

**multiplier *value***

ホールドダウン時間を 3 ～ 50 の数値として計算するために使用される値を指定します。

**使用上のガイドライン** エコーモード、および BFD パケット間の送信間隔などの BFD パラメータを指定します。

**例**

このインターフェイスに有効エコーモードを適用するには、次のコマンドを使用します。

```
bfd echo
```

次のコマンドは、BFD interval パラメータを設定します。

```
bfd interval 3000 min_rx 300 multiplier 3
```

# ciot-optimisation

このコマンドは、UE のコントロールプレーン（CP）の CIoT 最適化を設定するために使用されます。

**Command_Product** MME

**Command_Privilege** セキュリティ管理者、管理者

**コマンド モード** [Exec] > [Global Configuration] > [Call Control Profile Configuration]

**configure > call-control-profile** *profile_name*

上記のコマンドシーケンスを入力すると、次のプロンプトが表示されます。

```
[local]host_name(config-call-control-profile-profile_name)#
```

**構文の説明**

```
ciot-optimisation { cp-optimisation { access-type { all | nb-iot |
wb-eutran } |  ciot-capable-ue } | eps-attach-wo-pdn access-type { all
 | nb- iot | wb-eutran } }
remove ciot-optimisation cp-optimisation  ciot-capable-ue
remove ciot-optimisation eps-attach-wo-pdn access-type { all | nb-iot
| wb-eutran }
```

**remove**

キーワードの remove を指定すると、既存の設定が削除されます。

**cp-optimisation**

コントロールプレーンの最適化を UE に対して有効にするには、このキーワードを使用します。

**access-type**

このキーワードを使用してコントロールプレーンの最適化を有効にする必要があるアクセスタイプの拡張子を指定します。PDN を使用しないコントロールプレーンの最適化と EPS 接続は、NB-IoT と WB-EUTRAN RAT の両方またはいずれかで有効にできます。

**ciot-capable-ue**

CP を最適化するかどうかの決定には、ue-nw-capability のみを使用します。

**all**

このキーワードを使用して RAT タイプの WB-EUTRAN と NB-IOT の両方でコントロールプレーンの最適化を有効にします。このキーワードを使用すると、オペレータは簡単に設定できます。NB-IoT と WB-EUTRAN は両方とも、すべての機能に対する 2 つの独立したアクセスタイプと見なされます。

**nb-iot**

このキーワードを使用して RAT タイプ NB-IoT でコントロールプレーンの最適化を有効にします。

**wb-eutran**

このキーワードを使用して RAT タイプ WB-EUTRAN でコントロールプレーンの最適化を有効にします。

**eps-attach-wo-pdn**

このキーワードを使用して UE に対する PDN サポートなしに EPS 接続を有効にします。

## 使用上のガイドライン

このコマンドを使用して RAT タイプでコントロールプレーンの最適化を設定し、また、UE に対する PDN サポートなしに EPS 接続を設定します。このコマンドはデフォルトでは無効になっています。call-control-profile は、operator-policy または IME-TAC グループに関連付けることができます。そのため、サブスクライバ（IMSI）ごと、あるいはサブスクライバのグループまたは IMEI グループごとに CIoT の最適化を有効にすることも、無効にすることもできます。CIoT の最適化は、NB-IoT と WB-EUTRAN の両方の RAT またはいずれかで有効にできます。1 つの RAT タイプを有効にしても、別の RAT タイプは無効になりません。

### 例

次のコマンドを使用し、アクセスタイプを NB-IoT に指定することによって、コントロールプレーンの最適化を設定します。

```
ciot-optimisation cp-optimisation access-type nb-iot
```

次のコマンドを使用し、UE に対する PDN サポートを使用せずに EPS 接続を設定するには、アクセスタイプとして WB-EUTRAN を指定します。

**ciot-optimisation eps-attach-wo-pdn access-type wb-eutran**

# debug bfd

Bidirectional Forwarding Detection（BFD）を デバッグするためのデバッグオプションを有効または無効にします。ロギングが有効になっている場合は、ロギングシステムに結果が送信されます。

**Command_Product**

すべて

**Command_Privilege**

セキュリティ管理者、管理者、オペレータ

**コマンド モード**

Exec

次のプロンプトが Exec モードで表示されます。

`[local]host_name#`

**構文の説明**

**[ no ] debug bfd [all | events ipc-error | ipc-events | nsm | packet | session]**
**[level-1 | level-2 | level-3]**

**no**

指定した IP インターフェイスや機能に対して IP デバッグを無効にすることを示します。

**bfd *name***

デバッグする IP インターフェイス/機能を指定します。

**all**：すべての BFD アイテムのデバッグを有効にします。

**events**：BFD イベントのデバッグを有効にします。

**ipc-error**：BFD プロセス間通信（IPC）エラーのデバッグを有効にします。

**ipc-events**: BFD プロセス間通信（IPC）イベントのデバッグをイネーブルにします。

**nsm**：BFD ネットワーク サービス マネージャのメッセージのデバッグを有効にします。

**packet**：BFD パケットのデバッグを有効にします。

**session**：BFD セッションのデバッグを有効にします。

**level-1 | level-2 | level-3**

オプションで、debug コマンドによって提供される情報の量を指定します。

- レベル 1 のデバッグでは、エラー、警告、およびクリティカルな 1 回限りのイベントが表示されます。デフォルトはレベル 1 です。

- レベル 2 のデバッグでは、エラー、警告、およびすべてのイベントが表示されます。
- レベル 3 のデバッグでは、エラー、警告、およびすべてのイベントがさらに詳細に表示されます。

**使用上のガイドライン** この **debug** コマンドは、BFD が有効になっている BGP ルータのネットワークの問題をトラブルシューティングするときに役立ちます。デバッグを停止するには、**no** キーワードを使用します。

**注意** このコマンドを発行すると、システムの設定やロードに応じて、システムのパフォーマンスに悪影響を及ぼす可能性があります。

**例**

次のコマンドは BFD のデバッグを有効または無効にします。

```
debug bfd
no debug bfd
```

# delay-tolerant-pdn

省電力モードで UE をサポートするように、PDN 接続の遅延許容動作を設定します。

**Command_Product** P-GW
S-GW
SAEGW

**Command_Privilege** セキュリティ管理者、管理者

**コマンド モード** [Exec] > [Global Configuration] > [Context Configuration] > [APN Configuration]

**configure** > **context** *context_name* > **apn** *apn_name*

上記のコマンドシーケンスを入力すると、次のプロンプトが表示されます。

`[context_name]host_name(config-apn)#`

**構文の説明**

```
delay-tolerant-pdn max-control-signal-buffer  1-4
no delay-tolerant-pdn
```

**no**

設定を削除し、デフォルト値に戻します。

**max-control-signal-buffer *1 ～ 4***

UE が省電力モード（PSM）である場合に、バッファされる P-GW が開始した制御シグナリングメッセージの最大数（1 ～ 4 の範囲）を設定します。

**使用上のガイドライン**

CLI が設定されている場合は、PDN が遅延許容動作をサポートしていることを示します。また、バッファできる制御信号の数は、**max-control-signal-buffer** で示されます。新しいルールが更新/作成ベアラーに送信されると、バッファされるトランザクションの数が 4 に制限されます。

デフォルトでは、コマンドは無効になっており、eDRX のサポートは適用されません。

この CLI コマンドは、新しいコールのセットアップ時、または S5/S8 インターフェイスへのハンドオフ手順中に有効になります。

**例**

次のコマンドは、UE が省電力モードのときに、*3* つの P-GW が開始した制御シグナリングメッセージをバッファするように設定します。

```
delay-tolerant-pdn max-control-signal-buffer  3
```

# diameter

このコマンドは、SCEF サービスの Diameter インターフェイスを設定します。

**Command_Product**

MME

**Command_Privilege**

管理者

**コマンド モード**

Exec > グローバル コンフィギュレーション > コンテキスト コンフィギュレーション > SCEF サービス コンフィギュレーション

**configure** > **context** *context_name* > **scef-service** *service_name*

上記のコマンドシーケンスを入力すると、次のプロンプトが表示されます。

```
[context_name]host_name(config-scef-service)#
```

**構文**

```
diameter { dictionary standard | endpoint endpoint_name }
[ no ] diameter endpoint
[ default ] diameter dictionary
```

**no**

プレフィックス no は、この設定を無効にします。

**default**

プレフィックスのデフォルトは、選択したパラメータのデフォルトの値を割り当てるか、または復元します。

**endpoint *endpoint_value***

このコマンドは、Diameter のエンドポイントを設定します。

*endpoint_name* は、1 ～ 63 文字の英数字文字列で表される Diameter サーバ用である必要があります。

**dictionary standard**

このコマンドは、インターフェイスに使用するディクショナリを設定します。上記の設定は、T6a Diameter インターフェイスでの SCEF を介した非 IP データの転送を設定するために使用できます。

**使用上のガイドライン**

このコマンドを使用して、Diameterインターフェイスを設定します。上記のコマンドを使用すると、T6a Diameter インターフェイスでの SCEF を介した非 IP データの転送を設定できます。

Diameterのエンドポイント名を指定する必要があります。システムにアクティブなコールがある場合は、Diameterのエンドポイントを削除することは推奨されません。そのため、「Method of Procedure」に従ってエンドポイントを削除してください。そうしないと、システムの動作が未定義になります。

**例**

次のコマンドは、エンドポイントに t6a-endpoint を使用して Diameter を設定します。

```
diameter endpoint t6a-endpoint
```

**例**

次のコマンドは、Diameter の標準ディクショナリを設定します。

```
diameter dictionary standard
```

# edrx

このコマンドは、Extended Discontinuous Reception（eDRX）を有効にし、MME 上でそれぞれのパラメータを設定します。

**Command_Product**

MME

**Command_Privilege**

管理者

**コマンド モード**

[Exec] > [Global Configuration] > [Call Control Profile Configuration]

**configure > call-control-profile** *profile_name*

上記のコマンドシーケンスを入力すると、次のプロンプトが表示されます。

```
[local]host_name(config-call-control-profile-profile_name)#
```

### 構文

```
edrx { ptw ptw_value edrx-cycle cycle_length_value | ue-requested } [
dl-buf-duration [ packet-count packet_count_value ] ]
remove edrx
```

#### remove

キーワード **remove** を指定すると、MME 上の eDRX 設定が無効になります。

#### ptw *ptw_value*

このキーワードは、PTW 値を設定するために使用します。

21.2 よりも前のリリース：*ptw_value* は「0」から「20」までの整数です。

21.2 以降のリリース：*ptw_value* は「0」から「15」までの整数です。

#### ue-requested

キーワード **ue-requested** は、受け入れる接続要求/TAU 要求メッセージ内の UE から受信した Paging Time Window（PTW）と eDRX サイクル長の UE 要求値を指定します。

#### edrx-cycle *cycle_length_value*

キーワード **edrx-cycle** は、eDRX サイクルの長さを設定するために使用します。*cycle_length_value* は、「512」から「262144」までの整数値です。これは、512 ～ 262144 までの 2 の倍数になります（512、1024、2048など）。

#### dl-buf-duration

キーワード **dl-buf-duration** は、UE をページングできない場合に DDN ACK 内でダウンリンクバッファ期間を送信するために使用します。

#### packet-count*packet_count_value*

キーワード **packet count** は、UE をページングできない場合に、DDN ACK 内で「DL Buffering Suggested Packet Count」を送信するために使用します。*packet_count_value* は、「0」から「65535」までの整数値です。*packet_count_value* がローカルに設定されていない場合、*packet_count_value* のサブスクリプションに指定された値が使用されます。サブスクリプション値を「0」にすることができます。この場合、ローカルで設定されている場合でも、そのサブスクライバに対してパケットカウント IE は送信されません。

### 使用上のガイドライン

このコマンドを使用して、MME 上で eDRX を有効にします。このコマンドは、MME の eDRX 機能の一部として設定されています。これにより、UE は必要に応じてネットワークに接続できます。eDRX を使用すると、H-SFN 同期時間（UTC 時間）に基づき、分、時間、または日の

単位でデバイスが非アクティブか、またはスリープモードの状態のままになります。eDRX の H-SFN 同期時間は、MME サービスレベルで設定されます。H-SFN 同期の設定については、「*MME* サービス コンフィギュレーション モードのコマンド」の章を参照してください。このコマンドはデフォルトでは無効になっています。

**例**

次のコマンドを使用して、PTW と eDRX サイクル長を設定します。また、このコマンドは、推奨されるパケット数とともに、DDN ACK 内でダウンロードバッファ期間を送信するためにも使用されます。

**edrx ptw 10 edrx-cycle 512 dl-buf-duration packet-count 10**

# gtpc

このサービスに対して、GPRS トンネリングプロトコル制御（GTP-C）プレーンの設定を行います。

**Command_Product**

ePDG

MME

P-GW

S-GW

SAEGW

SaMOG

SGSN

**Command_Privilege**

管理者

**コマンド モード**

[Exec] > [Global Configuration] > [Context Configuration] > [eGTP Service Configuration]

**configure** > **context** *context_name* > **egtp-service** *service_name*

上記のコマンドシーケンスを入力すると、次のプロンプトが表示されます。

```
[context_name]host_name(config-egtp-service)#
```

**構文の説明**

**gtpc { allow-on-congestion { apn-name** *apn_name* **| arp** *priority_level* **} | bind { ipv4-address** *ipv4_address* **[ ipv6-address** *ipv6_address* **] | ipv6-address** *ipv6_address* **[ ipv4-address** *ipv4_address* **] } | command-messages { dual-ip-stack-support } | disable cause-source | echo-interval** *seconds* **[ dynamic [ smooth-factor** *multiplier* **] ] echo-max-retransmissoins** *number* **| echo-retransmission-timeout** *seconds* **| error-response-handling | peer-salvation | ip qos-dscp {** *forwarding_type* **| max-remote-restart-counter-change** *integer* **} | max-retransmissions** *num* **| node-feature { network-triggered-service-restoration | pgw-restart-notification } | path-failure detection-policy { echo | control-restart-counter-change | echo-restart-counter-change } | private-extension overcharge-protection | reject s2b-ho-no-context |**

**retransmission-timeout** *seconds* **| retransmission-timeout-ms** *milliseconds* **}**
**no gtpc { allow-on-congestion { apn-name** *apn_name* **| arp** *priority_level* **} |**
**bind { ipv4-address** *ipv4_address* [ ipv6-address *ipv6_address* ] **| ipv6-address**
*ipv6_address* **[ ipv4-address** *ipv4_address*] **} | command-messages {**
**dual-ip-stack-support } | disable cause-source | echo-interval |**
**error-response-handling | node-feature {**
**cellular-iotnetwork-triggered-service-restoration |**
**pgw-restart-notification } | path-failure detection-policy |**
**private-extension overcharge-protection | reject s2b-ho-no-context }**
**default gtpc disable cause-source |{ echo-interval |**
**echo-max-retransmissons | echo-retransmission-timeout disable**
**cause-source| ip qos-dscp | max-retransmissions | node-feature {**
**cellular-iot network-triggered-service-restoration |**
**pgw-restart-notification } | path-failure detection-policy |**
**retransmission-timeout | retransmission-timeout-ms }**

**no**

設定された GTP-C 設定を無効または削除します。

**default**

指定されたパラメータをデフォルト値にリセットします。

**allow-on-congestion { apn-name *apn_name* | arp *priority_level* }**

重要 ***P-GW、SAEGW、および S-GW のみ***. この機能を使用するには、有効な VoLTE ライセンスキーがインストールされている必要があります。ライセンスの入手方法の詳細については、シスコのアカウント担当者にお問い合わせください。

指定された APN/ARP の輻輳状態にあるコールの優先順位付け済みの処理を有効にします。

- 優先順位付け済みの APN/ARP の処理が有効になっていて、EGTP demux で CSReq で受信した APN/ARP が、設定された優先順位付け済みの APN/ARP 値のいずれかと一致する場合、輻輳制御のために有効な CSReq は EGTP demux で拒否されません。
- この機能は、新しい着信コールの CSReq 処理にのみ影響します。
- P-GW が開始した専用ベアラーの作成と更新は、この設定によって変更されません。

**apn-name** *apn_name*：輻輳が発生している場合でも、このアクセスポイント名（APN）へのコールを許可するようにゲートウェイを設定します。*apn_name* は、1 ～ 64 文字の英数字の文字列です。最大 3 つの Apn を設定できます。

**arp** *priority_level*：輻輳が発生している場合でも、この ARP のコールを許可するようにゲートウェイを設定します。*priority_level* は、優先順位の値を 1 ～ 15 の整数で設定します。最大 8 つの ARP 値を設定できます。

重要 APN から ARP へのマッピングはありません。

**bind { ipv4-address *ipv4_address* [ ipv6-address *ipv6_address* ] | ipv6-address *ipv6_address* [ ipv4-address *iv4p_address* ] }**

IPv4 アドレス、IPv6 アドレス、またはその両方を使用して、サービスをインターフェイスにバインドします。

**ipv4-address** *ipv4_address* **[ ipv6-address** *ipv6_address* **]**：このサービスを設定されたインターフェイスの IPv4 アドレスにバインドします。必要に応じて、IPv6 アドレスを使用して設定されたインターフェイスにサービスをバインドします。

*ipv4_address* は、IPv4 ドット付き 10 進表記を使用して入力する必要があります。

*ipv6_address* は、IPv6 コロン区切り 16 進表記を使用して入力する必要があります。

**ipv6-address** *ipv6_address* **[ ipv4-address** *ipv4_address* **]**：このサービスを設定されたインターフェイスの IPv6 アドレスにバインドします。必要に応じて、IPv4 アドレスを使用して設定されたインターフェイスにサービスをバインドします。

*ipv6_address* は、IPv6 コロン区切り 16 進表記を使用して入力する必要があります。

*ipv4_address* は、IPv4 ドット付き 10 進表記を使用して入力する必要があります。

重要 S2b インターフェイスでのバインド GTP-C サービスの場合は、IPv6 バインドアドレスまたは IPv4 バインドアドレスを使用する必要があります。IPv4 アドレスと IPv6 アドレスの両方をバインドすることは、ePDG ではサポートされていません。

**ipv6-address***Ipv6_address* **[ ipv4-address** *ipv4_address* **]** オプションは、現在 SGSN ではサポートされていません。

**cellular-iot**

eGTP サービスでサポートされているセルラー IoT 機能を有効にします。

**command-messages dual-ip-stack-support**

**command-messages**：S-GW および P-GW. での MBC/DBC/BRC メッセージに関連する設定。

**dual-ip-stack-support**：サポートされている場合は、IPv4/IPv6 トランスポートの両方でコマンドメッセージを処理できるようにします。デフォルトでは、機能は有効です。

**disable cause-source**

**disable**：eGTPC レベルで機能を無効にします。

**cause-source**：Cause IE で要因となっているビットを無効にします。

**echo-interval *seconds* [ dynamic [ smooth-factor *multiplier* ] ]**

エコー要求メッセージを送信するまでの時間（秒単位）を設定します。*seconds* は、60 ～ 3600 の整数です。

デフォルト：60

**dynamic**：eGTP サービスのダイナミックエコータイマーを有効にします。ダイナミックエコータイマーは、計算されたラウンドトリップタイマー（RTT）を使用して、ピアノードへの異なるパスでのバリアンスをサポートします。

**smooth-factor** *multiplier*：ダイナミックエコータイマーに乗数を導入します。乗数は 1 ～ 5 の整数です。

デフォルト：2

**max-remote-restart-counter-change整数**

P-GW がピアの再起動を検出するまでのカウンタの変更を指定します。新しい再起動カウンタと古い再起動カウンタの絶対差が設定された値よりも小さい場合にのみ、ピアの再起動が検出されることに注意してください。たとえば、**max-remote-restart-counter-change** が10で、現在のピアの再起動カウンタが 251 の場合は、新しい再起動カウンタが 252 ～ 255 または 0 ～ 5 の場合にのみ、eGTP はピアの再起動を検出します。同様に、保存された再起動カウンタが 1 の場合は、新しい再起動カウンタが 2 ～ 11 の場合にのみ、eGTP はピアの再起動を検出します。

有効な設定値は、1 ～ 255 です。

推奨設定値は 32 です。

デフォルト設定値は 255 です。

**echo-max-retransmissions *number***

GTP エコー要求の再試行の最大回数を設定します。*number*は 0 ～ 15 の整数です。
**echo-max-retransmissions** オプションが設定されていない場合、エコーの再試行の最大回数に **max-retransmissions** が設定されます。

デフォルト：4

**echo-retransmission-timeout *seconds***

eGTP サービスのエコー再送信タイムアウトを秒単位で設定します。*seconds*は、1 ～ 20 の範囲の整数です。

ダイナミックエコーが有効になっている場合（**gtpc echo-interval dynamic**）、このコマンドで設定した値が動的最小値として機能します（スムース係数を乗じた RTT がこのコマンドで設定された値よりも小さい場合、サービスはこの値を使用します）。

デフォルト：3

**error-response-handling**

S-GW で error-response-handling を有効にします。このコマンドが eGTP サービスで有効になっている場合、検証中にメッセージをドロップするのではなく、ピアから不正な応答を受信する

と、 eGTP-C は S-GW に不正な応答を受信したことを通知します。S-GW は、不正な応答を受信したという eGTP サービスからのこの通知を利用して、他のピアに適切な応答を送信します。

**peer-salvation**

このコンテキストで、EGTP サービスの非アクティブな GTPv2 ピアの peer-salvation を有効にします。有効にすると、この機能は特定の egtp-service レベルで有効になります。

この機能が egtp-service レベルで有効になっている場合は、コンテキストレベルで有効にする必要があります。設定順序は、この機能を有効にすることには依存しません。

peer-salvation が有効になっている場合は、コンテキストレベルで設定されたパラメータが使用されます。製品のすべての設定済みサービスに、peer-salvation ガ設定されていることを確めます。たとえば、sgw-services（egtp-service）です。

（注）

- peer-salvation が有効になっている場合は、コンテキストレベルで設定されたパラメータが使用されます。製品のすべての設定済みサービスに、peer-salvation ガ設定されていることを確めます。たとえば、sgw-services（egtp-service）です。
- 特定のピアのすべての情報（ピア統計情報やリカバリカウンタなど）は、回収後に失われます。
- コンテキストレベルの設定は、egtpinmgr と egtpegmgr 個別に適用されます。

**ip qos-dscp { *forwarding_type* }**

LTE コンポーネントから発信されたシグナリングパケットの外部ヘッダーでマークされる IP QoS DSCP のホップ単位の動作（PHB）を指定します。これは、標準規格ベースの機能（RFC 2597 および RFC 2474）です。

次の CS（クラスセレクタ）モードのオプションは、一部のネットワークデバイスで使用される IP 優先順位フィールドとの下位互換性をサポートするために提供されていることに注意してください。CS は IP 優先順位に 1 対 1 でマッピングします。CS1 では IP 優先順位の値が 1 です。IP 優先順位マーキングを使用した非 DSCP 認識ルータからパケットを受信した場合でも、DSCP ルータはエンコーディングをクラスセレクタのコードポイントとして認識できます。

次の転送タイプがサポートされています。

- **af11**：相対的優先転送 11 PHB の使用を指定します。

  これがデフォルトの設定です。

- **af12**：相対的優先転送 12 PHB の使用を指定します。
- **af13**：相対的優先転送 13 PHB の使用を指定します。
- **af21**：相対的優先転送 21 PHB の使用を指定します。
- **af22**：相対的優先転送 22 PHB の使用を指定します。

- **af23**：相対的優先転送 23 PHB の使用を指定します。
- **af31**：相対的優先転送 31 PHB の使用を指定します。
- **af32**：相対的優先転送 32 PHB の使用を指定します。
- **af33**：相対的優先転送 33 PHB の使用を指定します。
- **af41**：相対的優先転送 41 PHB の使用を指定します。
- **af42**：相対的優先転送 42 PHB の使用を指定します。
- **af43**：相対的優先転送 43 PHB の使用を指定します。
- **be**：ベストエフォートの転送 PHB の使用を指定します。
- **cs1**：クラスセレクタのコードポイント「CS1」の使用を指定します。
- **cs2**：クラスセレクタのコードポイント「CS2」の使用を指定します。
- **cs3**：クラスセレクタのコードポイント「CS3」の使用を指定します。
- **cs4**：クラスセレクタのコードポイント「CS4」の使用を指定します。
- **cs5**：クラスセレクタのコードポイント「CS5」の使用を指定します。
- **cs6**：クラスセレクタのコードポイント「CS6」の使用を指定します。
- **cs7**：クラスセレクタのコードポイント「CS7」の使用を指定します。
- **ef**：通常、低損失トラフィックと低遅延トラフィック専用の Expedited Forwarding（EF; 完全優先転送） PHB の使用を指定します。

次の表に、相対的優先転送の動作グループを示します。

| | クラス 1 | クラス 2 | クラス 3 | クラス 4 |
|---|---|---|---|---|
| 低廃棄確率 | AF11 | AF21 | AF31 | AF41 |
| 中廃棄確率 | AF12 | AF22 | AF32 | AF42 |
| 高廃棄確率 | AF13 | AF23 | AF33 | AF43 |

高いクラスでマークされたトラフィックは、輻輳期間中に優先されます。同じクラスのトラフィックに輻輳が発生した場合は、AF 値が高いパケットが最初にドロップされます。

**max-retransmissions *num***

パケットの再試行の最大回数を 0 ～ 15 の整数で設定します。

最大再送信回数に達すると、パスは失敗したと見なされます。

デフォルト：4

**node-feature pgw-restart-notification**

P-GW 再起動通知機能を有効にします。ノードは、設定が追加されるとすぐに、エコー内のピアノードに対してサポートされている新しい機能のアナウンスを開始します。

リリース17.0 以降では、S4-SGSN および MME は、P-GW 再起動通知（PRN）の受信とアドバタイジングをサポートしています。S4-SGSN や MME が eGTPC エコーの要求/応答メッセージで PRN の受信とアドバタイジングをサポートしていることを S-GW に通知するには、このコマンドオプションを設定する必要があります。

デフォルト：[Disabled]

**node-feature network-triggered-service-restoration**

このキーワードは、MME および S-GW のみに適用されます。

この eGTP サービスの 3GPP TS 23.007 リリース 11 に従って、Network Triggered Service Restoration（NTSR）機能を有効にします。

IMSI を含むダウンリンクデータ通知（DDN）メッセージを受信すると、MME は要求を受け入れ、UE に再接続を強制するために IMSI を含むページングを開始します。IMSI ベースの DDN 要求には、ゼロ TEID が含まれています。UE は接続されていないため、UE は MME のカバレッジエリア全体にページングされます。

接続要求に処理するために、異なる MME が eNodeB によって選択されている場合があります。DDN を処理した MME は、UE が 接続要求で応答したことを認識しないため、タイムアウト時にページングを停止します。

**path-failure detection-policy echo**

ピアへのエコータイムアウトによって検出されたパス障害時のセッションのクリーンアップを有効にします。

デフォルト：有効

無効になっている場合、ピアへのエコータイムアウトによって検出されたパス障害時にセッションのクリーンアップは行われません。ただし、SNMP トラップと SNMP ログは引き続きパス障害を示します。

**path-failure detection-policy control-restart-counter-change**

エコー要求/エコー応答メッセージの再起動カウンタが変更された場合に、パス障害検出ポリシーを有効にします。**max-remote-restart-counter-change** コマンドと組み合わせて使用します。

**path-failure detection-policy echo-restart-counter-change**

制御要求/制御応答メッセージの再起動カウンタが変更された場合に、パス障害検出ポリシーを有効にします。**max-remote-restart-counter-change** コマンドと組み合わせて使用します。

**private-extension overcharge-protection**

重要 StarOS 19.0 以降のリリースでは、このコマンドは廃止されています。

重要 過充電保護を使用するには、有効なライセンスキーがインストールされている必要があります。ライセンスの入手方法の詳細については、シスコのアカウント担当者にお問い合わせください。

Indication 情報要素またはプライベート拡張に、PDU に過充電保護関連のデータを含めるかどうかを制御します。

- このキーワードが eGTP サービスで有効になっている場合、eGTP-C は Indication IE ではなくプライベート拡張において、過充電保護関連のデータをエンコード/デコードします。
- このオプションが eGTP サービスで無効になっている場合、eGTP-C レイヤは、Indication IE で過充電保護関連のデータをエンコード/デコードします。
- デフォルトでは、このオプションは無効になっています。

**reject s2b-ho-no-context**

eGTP-C に UE コンテキストにない場合でも、S2b インターフェイスでのハンドオフコールを許可します。

**retransmission-timeout *seconds***

重要 17.3 以降のリリースでは、このオプションは廃止されました。**retransmission-timeout-ms** オプションを使用します。

GTPv2 制御パケット（エコーなし）の再送信のタイムアウト（秒単位）を 1 ～ 20 の整数で設定します。

デフォルト：5

**retransmission-timeout-ms *milliseconds***

GTP の制御パケットの再送信タイムアウトを、1000 から 20000 までの範囲のミリ秒単位（100 ステップ）で設定します。

デフォルト：5000

## 使用上のガイドライン

このコマンドを使用して、現在のサービスの GTP-C 設定を行います。

このインターフェイスは、S-GW または MME 上の S11 参照点の特性を前提としています。

S4-SGSN と LTE S-GW の間の通信では、インターフェイスは S4-SGSN 上の S4 参照点の特性を前提としています。S4-SGSN 上で **gtpc** コマンドを使用する前に、コンテキスト コンフィギュレーション モードで **egtp-service** コマンドを使用して、新規または既存のサービスを作成または入力する必要があります。eGTP サービスを設定したら、*SGSN* サービス コンフィギュレーション モードや *GPRS* サービス コンフィギュレーション モードで **associate** コマンドを使用して、S4-SGSN 上の設定済みの 2G サービスや 3G サービスにサービスを関連付ける必要があります。

**重要** このコマンドを変更すると、親サービス（eGTP/GTP-U サービスが設定されているサービス）が自動的に再起動します。サービスを再起動すると、親サービスに関連付けられているアクティブコールがドロップされます。

**重要** ePDG が IPv6 を S2b インターフェイスのトランスポートとしてサポートするため、ePDG では IPv6 バインドアドレスを使用する必要があります。

**例**

次のコマンドは、*112.104.215.177* の IPv4 アドレスを持つ GTP-C インターフェイスにサービスをバインドします。

```
gtpc bind ipv4-address 112.104.215.177
```

# gtpp attribute

オプション属性の指定を、GPRS/PDN/UMTS アクセスゲートウェイ が生成するコール詳細レコード（CDR）に表示できるようにします。また、属性フィールド値をエンコードすることによって、情報が CDR にどのように表示されるかを定義します。

**Command_Product**

GGSN

SGSN

P-GW

SAEGW

**Command_Privilege**

セキュリティ管理者、管理者

**コマンド モード**

[Exec] > [Global Configuration] > [Context Configuration]

**configure > context** *context_name*

上記のコマンドシーケンスを入力すると、次のプロンプトが表示されます。

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

構文の説明

**gtpp attribute { apn-ambr [ include-for-all-bearers | include-for-default-bearer | include-for-non-gbr-bearers ] | apn-ni | apn-selection-mode | charging-characteristic-selection-mode | camel-info | cell-plmn-id | { ciot-cp-optind | ciot-unipdu-cponly } | diagnostics [ abnormal-release-cause ] | direct-tunnel | duration-ms | dynamic-flag | dynamic-flag-extension | furnish-charging-information | imei | imsi-unauthenticated-flag | lapi last-ms-timezone | last-uli | local-record-sequence-number | losdv | ms-timezone | msisdn | node-id | node-id-suffix *STRING* | pdn-connection-id | pdp-address | pdp-type | pgw-ipv6-addr | pgw-plmn-id | plmn-id | qos max-length | rat | recordextension | record-extensions rat | record-type { sgsnpdprecord | sgwrecord } | served-mnai | served-pdp-pdn-address-extension | served-pdp-pdn-address-prefix-length | sgsn-change | sms { destination-number | recording-entity | service-centre } | sgw-ipv6-addr | sna-ipv6-addr | sponsor-id | start-time | stop-time | twanuli | uli | user-csg-information } +**

**default gtpp attribute { apn-ambr [ include-for-all-bearers | include-for-default-bearer | include-for-non-gbr-bearers ] | apn-ni | apn-selection-mode | charging-characteristic-selection-mode | camel-info | cell-plmn-id | { ciot-cp-optind | ciot-unipdu-cponly } | diagnostics [ abnormal-release-cause ] | direct-tunnel | duration-ms | dynamic-flag | dynamic-flag-extension | furnish-charging-information | imei | imsi-unauthenticated-flag | lapi last-ms-timezone | last-uli | local-record-sequence-number | losdv | ms-timezone | msisdn | node-id | node-id-suffix *STRING* | pdn-connection-id | pdp-address | pdp-type | pgw-ipv6-addr | pgw-plmn-id | plmn-id | qos max-length | rat | recordextension | record-extensions rat | record-type { sgsnpdprecord | sgwrecord } | served-mnai | served-pdp-pdn-address-extension | served-pdp-pdn-address-prefix-length | sgsn-change | sms { destination-number | recording-entity | service-centre } | sgw-ipv6-addr | sna-ipv6-addr | sponsor-id | start-time | stop-time | twanuli | uli | user-csg-information } +**

**no gtpp attribute { apn-ambr [ include-for-all-bearers | include-for-default-bearer | include-for-non-gbr-bearers ] | apn-ni | apn-selection-mode | charging-characteristic-selection-mode | camel-info | cell-plmn-id | { ciot-cp-optind | ciot-unipdu-cponly } | diagnostics [ abnormal-release-cause ] | direct-tunnel | duration-ms | dynamic-flag | dynamic-flag-extension | furnish-charging-information | imei | imsi-unauthenticated-flag | lapi last-ms-timezone | last-uli | local-record-sequence-number | losdv | ms-timezone | msisdn | node-id | node-id-suffix *STRING* | pdn-connection-id | pdp-address | pdp-type | pgw-ipv6-addr | pgw-plmn-id | plmn-id | qos max-length | rat | recordextension | record-extensions rat | record-type { sgsnpdprecord | sgwrecord } | served-mnai | served-pdp-pdn-address-extension | served-pdp-pdn-address-prefix-length | sgsn-change | sms { destination-number | recording-entity | service-centre } | sgw-ipv6-addr | sna-ipv6-addr | sponsor-id | start-time | stop-time | twanuli | uli | user-csg-information } +**

**default**

生成された CDR のデフォルトの GTPP 属性を設定します。また、生成された CDR の属性値のデフォルトの表示も設定します。

**no**

CDR から設定された GTPP 属性を削除します。

**apn-ambr [ include-for-all-bearers | include-for-default-bearer | include-for-non-gbr-bearers ]**

デフォルト：無効

このキーワードは、custom24 GTPP ディクショナリの PGW-CDR へのオプションフィールド「apn-ambr」の挿入を制御します。

重要 このキーワードオプションは、有効なライセンスがインストールされている場合にのみ使用できます。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。

APN集約最大ビットレート（AMBR）は、APNごとに保存されるサブスクリプションパラメータです。これにより、すべてのGBR以外のベアラーと同じAPNのすべてのPDN接続にわたって提供されることが予想される集約ビットレートが制限されます。これらの GBR 以外のベアラーのそれぞれがAPN AMBR全体を利用する可能性があります。たとえば、他のGBR以外のベアラーがトラフィックを伝送しない場合などです。APN AMBR は QoS 情報の一部として存在します。

15.0 以降のリリースでは、この CLI コマンドはすべての GTPP ディクショナリの SGW-CDR での APN-AMBR レポートをサポートするために、次の追加オプションとともに設定する必要があります。

- **include-for-all-bearers**：すべてのベアラー（GBR と GBR 以外）の SGW-CDR に APN-AMBR 情報を含めます。
- **include-for-default-bearer**：デフォルトのベアラーの APN-AMBR 情報を SGW-CDR に含めます。
- **include-for-non-gbr-bearers**：GBR 以外のベアラーの APN-AMBR 情報を含めます。

この機能は、CDR の事後処理を有効にし、請求システムに対する MVNO サブスクライバの実際の QoS 確認するために必要です。

重要 この CLI コマンドおよび関連付けられているオプションは、S-GW と P-GW 以外の製品では使用できません。オプション「**non-gbr-bearers-only**」は、S-GW と P-GW で使用できますが、その他のオプションは、S-GW でのみ使用できます。

P-GW の実装では、CLI コマンド「**gtpp attribute apn-ambr**」が設定されている場合、「**gtpp attribute apn-ambr non-gbr-bearers-only**」として処理されます。S-GW/P-GW コンボの場合、オプションのいずれかが設定されていると、その属性が使用可能であると見なされます。

**apn-ni**

デフォルト：有効

このキーワードは、CDR へのオプションフィールド「APN」の挿入を制御します。

**apn-selection-mode**

デフォルト：有効

このキーワードは、x-CDR へのオプションフィールド「APN Selection Mode」の挿入を制御します。

**camel-info**

SGSN のみ

SGSN CDR に CAMEL 固有のフィールドを含めるには、このキーワードを入力します。デフォルト：無効

**cell-plmn-id**

SGSN のみ

システムが M-CDR に Cell PLMN ID フィールドを挿入できるようにするには、このキーワードを入力します。デフォルト：無効

**charging-characteristic-selection-mode**

デフォルト：有効

このキーワードは、x-CDR へのオプションフィールド「Charging Characteristic Selection Mode」の挿入を制御します。

**ciot-cp-optind**

CDR にオプションフィールド「CP CIoT EPS optimisation indicator」を挿入します。

**ciot-unipdu-cponly**

CDR にオプションフィールド「UNI PDU CP Only Flag」を挿入します。

**diagnostics [ abnormal-release-cause ]**

デフォルト：無効

PDP コンテキストが解放されたときに作成される CDR にシステムが Diagnostic フィールドを挿入できるようにします。このフィールドには、次のいずれかの値が含まれます。

- **26**：GGSN の場合：GGSN が他の何らかの理由で「delete PDP context request」を送信した場合（オペレータが GGSN に「clear subscribers」と入力したなど）。SGSN の場合：2 番目の PDP コンテキストのアクティブ化要求または PDP コンテキスト変更要求がリソース不足のために拒否されたことを示すために、SGSN が S-CDR にこの原因コードを挿入します。

- **36**：GGSN：SGSN が GGSN に「delete PDP context request」を送信したために、PDP コンテキストが GGSN で非アクティブになっていることを示すため、G-CDR 内でこの原因コードが送信されます。SGSN の場合、この原因コードは、通常の MS またはネットワークによって開始された PDP コンテキストの非アクティブ化を示すために使用されます。
- **37**：ネットワークが QoS の変更を開始すると、SGSN は S-CDR 内で、MS によって開始された非アクティブ化要求メッセージが原因として受け入れられない QoS として拒否されたことを示すために送信されます。
- **38**：SGSN での GTP-C/GTP-U エコーのタイムアウトにより GGSN が「delete PDP context request」を送信した場合。SGSN がこの原因コードを送信した場合、パス障害、特に GTP-C/GTP-U のエコータイムアウトのために PDP コンテキストが非アクティブ化されていることを示します。
- **39**：SGSN only：このコードは、GGSN の再起動後に、ネットワーク（GGSN）が PDP コンテキストの再アクティブ化を要求したことを示します。
- **40**：RADIUS 接続解除要求メッセージの受信により、GGSN が「delete PDP context request」を送信した場合。

**abnormal-release-cause**：このキーワードは、SGW-CDR の診断フィールドへのベアラーの異常終了情報の挿入を制御します。CLI コマンド「**gtpp attribute diagnostics**」は、**abnormal-release-cause** を無効にして **diagnostics** フィールドを有効にすることに注意してください。**no gtpp attribute diagnostics** コマンドは、**abnormal-release-cause** と **diagnostics** フィールドの両方を無効にします。

**重要** ベアラーの異常終了機能は現在、custom34 と custom35 の GTPP ディクショナリに対してのみ適用できます。つまり、レコード終了の原因が「Abnormal Release」の場合、custom34 ディクショナリと custom35 ディクショナリでは SGW-CDR に、custom35 GTPP ディクショナリでは PGW-CDR にベアラー終了の原因が挿入されます。

**direct-tunnel**

デフォルト：無効

PGW-CDR/eG-CDR の直接トンネルフィールドを挿入します。

このキーワードは、GGSN、P-GW、および S-GW にのみ適用できます。

**duration-ms**

必須の Duration フィールドに含まれる情報が秒単位ではなくミリ秒単位で報告されることを指定します（標準で求められている場合）。デフォルト：無効

**dynamic-flag**

デフォルト：有効

このキーワードは、x-CDR へのオプションフィールド「Dynamic Flag」の挿入を制御します。

**dynamic-flag-extension**

デフォルト：有効

このキーワードは、x-CDR へのオプションフィールド「Dynamic Address Flag Extension」の挿入を制御します。

このフィールドは、デュアル PDP コンテキストに IPv4 アドレスが動的に割り当てられている場合に、CDR に表示されます。この拡張機能フィールドは、デュアルスタックベアラーのサポートが使用可能になるように、3GPP リリース 10 に準拠した CDR で必要です。

**furnish-charging-information**

デフォルト：無効

このキーワードは、eG-CDR と PGW-CDR へのオプションフィールド「pSFurnishChargingInformation」の挿入を制御します。

重要 Furnish Charging Information（FCI）機能は、custom43 ディクショナリを除き、3GPP Rel.7 と 3GPP Rel.8 に準拠したすべての GTPP ディクショナリに適用されます。このキーワードオプションは、有効なライセンスがインストールされている場合にのみ使用できます。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。

GTPP サーバ グループ コンフィギュレーション モードで **gtpp attribute furnish-charging-information** コマンドを使用するなど、コマンドレベルで有効になっている場合にのみ、PGW-CDR と eG-CDR には FCI が含まれます。

FCI が変更されるたびに、新しい Free-Format-Data（FFD）の値は既存の FFD に付加されるか、または Append-Free-Format-Data（AFFD）フラグに応じて既存の FDD 上で上書きされます。CDR は FCI の変更時には生成されません。

FCI は、メイン CDR と LOSDV でもサポートされています。トリガー（ボリューム、時間、RAT など）が発生するたびに、コマンドレベルで現在使用可能な FFD が両方の CDR の本文に追加されます。コマンドレベルで次の Credit-Control-Answer メッセージによって付加または上書きされるまで、コマンドレベルの同じ FFD が次の CDR の本文に追加されます。

custom43 ディクショナリの場合、FCI の実装は次のようになります。

- FCI が変更されるたびに、PGW-CDR は CDR を生成します。つまり、古いバケットをクローズし、生成された CDR 内に古い FCI の詳細が保持されます。
- CDR 内の PS-Free-Format-Data の変換は、ASCII 形式の 16 進数値（0 ～ 9 の数値）から整数の 10 進値への変換です。
- PS-Append-Free-Format-Data は常に上書きします。

**imei**

デフォルト：無効

SGSN の場合：IMEI 値を S-CDR に挿入します。

GGSN の場合：IMEISV 値を G-CDR に挿入します。

**imsi-unauthenticated-flag**

デフォルト：有効

このキーワードは x-CDR へのオプションフィールド「IMSI Unauthenticated Flag」の挿入を制御します。

サービス対象の IMSI が認証されていない場合、このフィールド「IMSI Unauthenticated Flag」は、設定されていれば、custom35 ディクショナリの P-GW CDR レコードに存在します。このフィールドは、3GPP TS 32.298 v10.7 に従って追加されます。

**lapi**

デフォルト：無効

CDR に Low Access Priority Indicator（LAPI）フィールドを挿入します。このフィールドは、MTC 機能のサポートに必要です。

UE が優先順位の低い接続を示している場合は、「lowPriorityIndicator」属性が CDR に挿入されます。

**last-ms-timezone**

デフォルト：[無効（Disabled）]

CDR フィールドに「Last MS-Timezone」を設定します。デフォルトのオプションが使用されている場合、このオプションは無効になります。

**last-uli**

デフォルト：無効

CDR フィールドに「Last ULI」を設定します。デフォルトのオプションが使用されている場合、このオプションは無効になります。

**local-record-sequence-number**

デフォルト：無効

このキーワードは、ローカルレコードシーケンス番号とノード ID の両方を提供します。x-CDR では、このフィールドはノードによって生成された CDR の数が示されます。また、このフィールドはセッションマネージャ内で一意です。

PDP コンテキストがリリースされた場合や、設定に基づいて CDR の一部が生成された場合などのいくつかの理由で、Node ID フィールドが x-CDR に挿入されます。このフィールドは、SGSN サービスや GGSN サービスの名前に自動的に付加された AAA マネージャ識別子から構成されます。

Node ID フィールドの最大長は 20 バイトであるため、SGSN サービスまたは GGSN サービスの名前が切り捨てられることがあります。各 AAA マネージャは単独で CDR を生成するため、

Local Record Sequence Number フィールドと Node ID フィールドは CDR を独自に識別できます。

重要 **gtpp single-source centralized-lrsn** が設定されている場合、「Node-ID」は指定した NodeID サフィックスのみで構成されます。NodeID サフィックスが設定されていない場合は、GTPP グループ名が使用されます。デフォルトの GTPP グループの場合、GTPP コンテキスト名が使用されます。**gtpp single-source centralized-lrsn** が設定されている場合は、Sessmgr によって生成された CDR のノード ID 形式は <1-byte Sessmgr restartvalue><3-byte Sessmgr instance number> <node-id-suffix> となります。**gtpp single-source centralized-lrsn** が設定されていない場合は、ACSmgr によって生成された CDR のノード ID 形式は <1-byte ACSmgr restart-value> <3-byte ACSmgr instance number> <Active charging service-name> となります。

**losdv**

デフォルト：有効

このキーワードは、x-CDR へのオプションフィールド「List of Service Data」の挿入を制御します。

**ms-timezone**

デフォルト：有効

このキーワードは、x-CDR へのオプションフィールド「MS-Timezone」の挿入を制御します。

**msisdn**

デフォルト：有効

このキーワードは、x-CDR へのオプションフィールド「MSISDN」の挿入を制御します。

**node-id**

デフォルト：有効

このキーワードは、x-CDR へのオプションフィールド「Node ID」の挿入を制御します。

**node-id-suffix 文字列**

デフォルト：無効

GTPP CDR の NodeID フィールドに使用するために設定した Node-ID-Suffix を 1 ～ 16 文字の英数値文字列として指定します。各セッションマネージャタスクは、GTPP コンテキストごとに一意の NodeID 文字列を生成します。

重要 NodeID フィールドは、*ndddSTRING* 形式の印刷可能な文字列です。*n*：最初の桁は、0 ～ 7 の値を持つ Sessmgr 再起動カウンタです。*add*：sessmgr インスタンスの数です。指定した NodeID-suffix をすべての CDR で使用します。「Node-ID」フィールドは、sessMgr リカバリカウンタ（1 桁）*n* + AAA マネージャ識別子（3 桁）*ddd* + 設定されたノード ID サフィックス（1 ~16 文字）の文字列で構成されます。一元化 LRSN 機能が有効になっている場合、「Node-ID」フィールドは、指定した NodeID-suffix のみで構成されます（NodeID-prefix は挿入されません）。このオプションが設定されていない場合は、代わりに GTPP グループ名が使用されます（デフォルトの GTPP グループの場合は context-name が使用されます）。

重要 この **node-id-suffix** が設定されていない場合、GGSN は GTPP コンテキスト名を Node-id-suffix を使用し（16 文字に切り捨てられます）、SGSN は GTPP グループ名を node-id-suffix として使用します。

**pdn-connection-id**

デフォルト：有効

このキーワードは、x-CDR へのオプションフィールド「PDN Connection ID」の挿入を制御します。

**pdp-address**

デフォルト：有効

このキーワードは、x-CDR へのオプションフィールド「PDP Address」の挿入を制御します。

**pdp-type**

デフォルト：有効

このキーワードは、x-CDR へのオプションフィールド「PDP Type」の挿入を制御します。

**pgw-ipv6-addr**

デフォルト：無効

このオプションを指定すると、P-GW IPv6 アドレスを設定できます。

重要 この属性は、custom24 と custom35 の SGW-CDR ディクショナリでの設定を制御できます。

**pgw-plmn-id**

デフォルト：有効

このキーワードは、x-CDR へのオプションフィールド「PGW PLMN-ID」の挿入を制御します。

**plmn-id [ unknown-use ]**

デフォルト：有効

SGSNの場合は、ディクショナリがサポートしている場合に提供されるS-CDR内のSGSN PLMN識別子の値（RAI）を報告します。

GGSN では、SGSN によって GTP の PDP コンテキスト作成要求でもともと提供されていた G-CDR 内で SGSN PLMN 識別子の値（RAI）を報告します。SGSN が提供していない場合は、省略されます。

通常、SGSN PLMN-id 情報が使用できない場合、属性 sgsnPLMNIdentifier は CDR に挿入されません。このキーワードを使用すると、SGSN PLMN-id が使用できない場合に、特定の値を持つ sgsnPLMNIdentifier を挿入できます。

**unknown-use** *hex_num*：PLMN-id を提供していない外部 SGSN を識別する 0x0 ～ 0xFFFFFF の 16 進数です（GGSN のみ）。

**qos max-length**

デフォルト：無効

このオプションを指定すると、S-CDR と SaMOG CDR 内で送信された QoS に関連するパラメータが変更されます。この **max-length** オプションは、CDR 内で送信される QoS の長さを変更するために使用されます。**qos_value** は 4 ～ 24 の整数である必要があります。

この機能は、Rel.7+ QoS の形式をサポートするために導入されました。

**rat**

デフォルト：有効

SGSNの場合：S-CDR内のRAT（無線アクセステクノロジータイプを識別する）値を挿入します。

GGSN の場合：G-CDR 内の RAT（無線アクセステクノロジータイプを識別する）値が含まれます。

**recordextension**

デフォルト：無効

このキーワードは、x-CDR へのオプションフィールド「RecordExtension」の挿入を制御します。

**record-extensions rat**

デフォルト：無効

ネットワークオペレータやメーカーが、3GPP TS 32.298 リリース 7 以降の標準レコードの定義に従って、CDR に独自の推奨拡張を追加できるようにします。

**record-type { sgsnpdprecord | sgwrecord }**

重要　このキーワードは、SaMOG 混合モードライセンス（3G と 4G の両方をサポート）が設定されている場合にのみ使用できます。

デフォルト：sgwrecord

使用する SaMOG CDR タイプを指定します。

SaMOG 3G ライセンスの場合、このキーワードは使用できません。ただし、デフォルトのレコードタイプとして sgsnpdprecord タイプが使用されます。

**served-mnai**

デフォルト：無効

このキーワードは、x-CDR へのオプションフィールド「Served MNAI」の挿入を制御します。

**served-pdp-pdn-address-extension**

デフォルト：無効

IPv4v6 デュアルスタック PDP アドレスタイプのサポートでは、このキーワードを指定すると、サービスが IPv4v6 アドレス情報を CDR に挿入します。IPv4 アドレスが Served PDP PDN Address Extension フィールドに挿入され、IPv6 アドレスが Served PDP Address フィールドか Served PDP PDN Address フィールドに挿入されます。

重要　この属性は、GTPP ディクショナリが custom34 に設定されている場合は表示されません。

（注）　SGSN の場合、**served-pdp-pdn-address-extension** を有効にした時点で、次のディクショナリを除くすべてのカスタム S-CDR ディクショナリが CDR フィールド「Served PDP/ PDN Address extension」をサポートします。

- custom17
- custom18
- custom23
- custom42
- custom41

**served-pdp-pdn-address-prefix-length**

デフォルト：有効

IPv6 プレフィックス委任のサポートでは、このキーワードを指定すると、サービスが x-CDR にこの「Served PDP PDN Address」フィールドを挿入します。

このフィールドが設定されている場合、servedPDPPDNAddress フィールドは、3GPP 32.298 に概説されているように IPv6 プレフィックス長を報告できます。プレフィックス長は、次の場合にのみ報告されます。

- 設定されている
- デフォルト長の 64 ではない
- IPv6 コールか IPv4v6 コールである

**sgsn-change**

デフォルト：有効

このキーワードは SGSN に固有であり、ライセンスが制限されています。

このキーワードは、S-CDR への S-CDR 属性「SGSN Change」の挿入を制御します。デフォルトで有効になっており、属性「SGSN Change」がデフォルトで S-CDR に挿入されます。

（注） SGSN 固有の custom33 ディクショナリの場合は、課金の問題を防ぐために、アップグレード前にこのキーワードを無効にすることを推奨します。

**sgw-ipv6-addr**

デフォルト：[Disabled]

このオプションを指定すると、S-GW IPv6 アドレスを設定できます。

重要 この属性は、custom24 と custom35 の SGW-CDR ディクショナリでの設定を制御できます。

**sms { destination-number | recording-entity | service-centre }**

このキーワードは SGSN に固有です。

このキーワードを入力すると、SMS-MO-CDR または SMS-MT-CDR に SMS 関連フィールドを挿入します。

**destination-number**：SMS-MO-CDR または SMS-MT-CDR に「destinationNumber」フィールドを挿入します。

**recording-entity**：SMS-MO-CDR または SMS-MT-CDR に「recordingEntity」フィールドを挿入します。

**service-centre**：SMS-MO-CDR または SMS-MT-CDR に「serviceCentre」フィールドを挿入します。

**sna-ipv6-addr**

デフォルト：[Disabled]

このオプションを指定すると、サービスノードの IPv6 アドレス（SNAv6）を設定できます。

重要　この属性は、custom24 と custom35 の SGW-CDR ディクショナリでの設定を制御できます。

**sponsor-id**

デフォルト：[Disabled]

PGW-CDR に Sponsor ID フィールドと Application-Service-Provider-Identity フィールドを挿入します。

PCEF がスポンサー提供データの接続機能をサポートしている場合は PGW-CDR に「Sponsor ID」属性と「Application-Service-Provider-Identity」属性が挿入されます。または、3GPP TS 29.212 で説明したとおり、必要なレポートレベルはスポンサー提供の接続になります。

この機能は、CDR についてのリリース 11 3GPP 仕様に準拠するように実装されます。したがって、リリース 11 に準拠する（custom35）すべての GTPP ディクショナリにこの動作が適用されます。

**start-time**

デフォルト：有効

このキーワードは、x-CDR へのオプションフィールドの「Start-Time」の挿入を制御します。

**stop-time**

デフォルト：有効

このキーワードは、x-CDR へのオプションフィールドの「Stop-Time」の挿入を制御します。

**twanuli**

デフォルト：[Disabled]

このキーワードは、CDR へのオプションフィールドの「TWAN User Location Information」の挿入を制御します。

**uli**

デフォルト：有効

このキーワードは、CDR にオプションフィールド「ユーザロケーション情報」を含めることを制御します。

**user-csg-information**

デフォルト：[Disabled]

このキーワードは、x-CDR へのオプションフィールド「User CSG Information」の挿入を制御します。

**重要** 現時点では、UCI 値は SGW-CDR でのみサポートされています。

この属性は、GTPP ディクショナリが custom11、custom34、または custom35 に設定されている場合は表示されません。

**+**

複数の属性を設定するために、このコマンドを複数回入力できることを示します。

**使用上のガイドライン** このコマンドを使用して、生成された CDR に含めるオプション情報フィールドのタイプを設定します（SGSN から M-CDR、S-CDR、S-SMO-CDR、S-SMT-CDR、および GGSN から G-CDR、eG-CDR）。これには、AGW（SGSN/GGSN/P-GW/SAEGW）を使用します。さらに、一部の必須フィールドの情報がどのように報告されるかを制御します。

標準ではオプションとして記載されており、上記に記載されていないフィールドは、レコードの拡張機能の場合を除き（存在することはない）、CDR に常に存在するようになります。

**重要** このコマンドは、複数の GTPP 属性を設定するために、さまざまなキーワードを使用して複数回繰り返すことができます。

**例**

次のコマンドは、CDR 内の Duration フィールドに指定された時間をミリ秒単位で報告するようにシステムを設定します。

```
gtpp attribute duration-ms
```

# gtpp attribute

GSN（GGSN または SGSN）が生成する CDR のオプションフィールドの一部の指定を有効にします。また、情報の表示方法を指定を有効にします。多くのキーワードは、S-GW と P-GW の CDR にも適用されます。

**Command_Product**

GGSN

P-GW

SAEGW

SaMOG

SGSN

S-GW

**Command_Privilege**

セキュリティ管理者、管理者

## コマンド モード

Exec > グローバル コンフィギュレーション > コンテキスト コンフィギュレーション > GTPP サーバ グループ コンフィギュレーション

**configure > context** *context_name* **> gtpp group** *group_name*

上記のコマンドシーケンスを入力すると、次のプロンプトが表示されます。

```
[context_name]host_name(config-gtpp-group)#
```

## 構文の説明

**gtpp attribute { apn-ambr [ include-for-all-bearers |
include-for-default-bearer | include-for-non-gbr-bearers ] | apn-ni |
apn-selection-mode | charging-characteristic-selection-mode | camel-info
 | cell-plmn-id |  { ciot-cp-optind | ciot-unipdu-cponly } | diagnostics[
 abnormal-release-cause ] | direct-tunnel | duration-ms | dynamic-flag
 | dynamic-flag-extension | extended-bitrate |
furnish-charging-information | imei | imsi-unauthenticated-flag | lapi
 last-ms-timezone | last-uli |
| local-record-sequence-number | losdv | ms-timezone | msisdn | node-id
 | node-id-suffix** *STRING* **packet-count | pco-nai | pdn-connection-id |
pdp-address | pdp-type | pgw-ipv6-addr | pgw-plmn-id | plmn-id | qos
max-length | rat | recordextension | record-extensions rat | record-type
 { sgsnpdprecord | sgwrecord } | served-mnai |
served-pdp-pdn-address-extension | served-pdp-pdn-address-prefix-length
 | sgsn-change |  sms { destination-number | recording-entity |
service-centre } | sgw-ipv6-addr | sna-ipv6-addr | sponsor-id |
start-time | stop-time | twanuli | ue-tun-ip-port | uwanuli | uli |
user-csg-information } +
default gtpp attribute { apn-ambr [ include-for-all-bearers |
include-for-default-bearer | include-for-non-gbr-bearers ] | apn-ni |
apn-selection-mode | charging-characteristic-selection-mode | camel-info
 | cell-plmn-id |  { ciot-cp-optind | ciot-unipdu-cponly } | diagnostics[
 abnormal-release-cause ] | direct-tunnel | duration-ms | dynamic-flag
 | dynamic-flag-extension | furnish-charging-information | imei |
imsi-unauthenticated-flag | lapi last-ms-timezone | last-uli |
| local-record-sequence-number | losdv | ms-timezone | msisdn | node-id
 | node-id-suffix** *STRING* **| pdn-connection-id | pdp-address | pdp-type |
 pgw-ipv6-addr | pgw-plmn-id | plmn-id | qos max-length | rat |
recordextension | record-extensions rat | record-type { sgsnpdprecord
| sgwrecord } | served-mnai | served-pdp-pdn-address-extension |
served-pdp-pdn-address-prefix-length | sgsn-change |  sms {
destination-number | recording-entity | service-centre } | sgw-ipv6-addr
 | sna-ipv6-addr | sponsor-id | start-time | stop-time | twanuli |
uwanuli | uli | user-csg-information } +
no gtpp attribute { apn-ambr [ include-for-all-bearers |
include-for-default-bearer | include-for-non-gbr-bearers ] | apn-ni |
apn-selection-mode | charging-characteristic-selection-mode | camel-info
 | cell-plmn-id |  { ciot-cp-optind | ciot-unipdu-cponly } | diagnostics[
 abnormal-release-cause ] | direct-tunnel | duration-ms | dynamic-flag
 | dynamic-flag-extension | extended-bitrate |
furnish-charging-information | imei | imsi-unauthenticated-flag | lapi
 last-ms-timezone | last-uli |
| local-record-sequence-number | losdv | ms-timezone | msisdn | node-id
 | node-id-suffix** *STRING* **packet-count | pco-nai | pdn-connection-id |**

**pdp-address | pdp-type | pgw-ipv6-addr | pgw-plmn-id | plmn-id | qos max-length | rat | recordextension | record-extensions rat | record-type { sgsnpdprecord | sgwrecord } | served-mnai | served-pdp-pdn-address-extension | served-pdp-pdn-address-prefix-length | af-record-info | sgsn-change | sms { destination-number | recording-entity | service-centre } | sgw-ipv6-addr | sna-ipv6-addr | sponsor-id | start-time | stop-time | twanuli | ue-tun-ip-port | uwanuli | uli | user-csg-information } +**

**default**

この GTPP グループ設定のデフォルトの属性値をリセットします。

**no**

生成されたCDRに情報が存在しないように、指定したオプションフィールドを無効にします。

**apn-ambr [ include-for-all-bearers | include-for-default-bearer | include-for-non-gbr-bearers ]**

デフォルト：無効

このキーワードは、custom24 GTPP ディクショナリの PGW-CDR へのオプションフィールド「apn-ambr」の挿入を制御します。

**重要** このキーワードオプションは、有効なライセンスがインストールされている場合にのみ使用できます。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。

APN集約最大ビットレート（AMBR）は、APN ごとに保存されるサブスクリプションパラメータです。これにより、すべてのGBR以外のベアラーと同じAPNのすべてのPDN接続にわたって提供されることが予想される集約ビットレートが制限されます。これらの GBR 以外のベアラーのそれぞれがAPN AMBR全体を利用する可能性があります。たとえば、他のGBR以外のベアラーがトラフィックを伝送しない場合などです。APN AMBR は QoS 情報の一部として存在します。

15.0以降のリリースでは、この CLI コマンドはすべての GTPP ディクショナリの SGW-CDR での APN-AMBR レポートをサポートするために、次の追加オプションとともに設定する必要があります。

- **include-for-all-bearers**：すべてのベアラー（GBR と GBR 以外）の SGW-CDR に APN-AMBR 情報を含めます。
- **include-for-default-bearer**：デフォルトのベアラーの APN-AMBR 情報を SGW-CDR に含めます。
- **include-for-non-gbr-bearers**：GBR 以外のベアラーの APN-AMBR 情報を含めます。

この機能は、CDR の事後処理を有効にし、請求システムに対する MVNO サブスクライバの実際の QoS 確認するために必要です。

**重要** この CLI コマンドおよび関連付けられているオプションは、S-GW と P-GW 以外の製品では使用できません。オプション「**non-gbr-bearers-only**」は、S-GW と P-GW で使用できますが、その他のオプションは、S-GW でのみ使用できます。

P-GW の実装では、CLI コマンド「**gtpp attribute apn-ambr**」が設定されている場合、「**gtpp attribute apn-ambr non-gbr-bearers-only**」として処理されます。S-GW/P-GW コンボの場合、オプションのいずれかが設定されていると、その属性が使用可能であると見なされます。

**apn-ni**

デフォルト：有効

このキーワードは、CDR へのオプションフィールド「APN」の挿入を制御します。

**apn-selection-mode**

デフォルト：有効

このキーワードは、x-CDR へのオプションフィールド「APN Selection Mode」の挿入を制御します。

**camel-info**

SGSN のみ

SGSN CDR に CAMEL 固有のフィールドを含めるには、このキーワードを入力します。デフォルト：無効

**cell-plmn-id**

SGSN のみ

システムが M-CDR に Cell PLMN ID フィールドを挿入できるようにするには、このキーワードを入力します。デフォルト：無効

**charging-characteristic-selection-mode**

デフォルト：有効

このキーワードは、x-CDR へのオプションフィールド「Charging Characteristic Selection Mode」の挿入を制御します。

**ciot-cp-optind**

CDR にオプションフィールド「CP CIoT EPS optimisation indicator」を挿入します。

**ciot-unipdu-cponly**

CDR にオプションフィールド「UNI PDU CP Only Flag」を挿入します。

**diagnostics [ abnormal-release-cause ]**

デフォルト：無効

PDP コンテキストが解放されたときに作成される CDR にシステムが Diagnostic フィールドを挿入できるようにします。このフィールドには、次のいずれかの値が含まれます。

- **26**：GGSN の場合：GGSN が他の何らかの理由で「delete PDP context request」を送信した場合（オペレータが GGSN に「clear subscribers」と入力したなど）。SGSN の場合：2 番目の PDP コンテキストのアクティブ化要求または PDP コンテキスト変更要求がリソース不足のために拒否されたことを示すために、SGSN が S-CDR にこの原因コードを挿入します。
- **36**：GGSN：SGSN が GGSN に「delete PDP context request」を送信したために、PDP コンテキストが GGSN で非アクティブになっていることを示すため、G-CDR 内でこの原因コードが送信されます。SGSN の場合、この原因コードは、通常の MS またはネットワークによって開始された PDP コンテキストの非アクティブ化を示すために使用されます。
- **37**：ネットワークが QoS の変更を開始すると、SGSN は S-CDR 内で、MS によって開始された非アクティブ化要求メッセージが原因として受け入れられない QoS として拒否されたことを示すために送信されます。
- **38**：SGSN での GTP-C/GTP-U エコーのタイムアウトにより GGSN が「delete PDP context request」を送信した場合。SGSN がこの原因コードを送信した場合、パス障害、特に GTP-C/GTP-U のエコータイムアウトのために PDP コンテキストが非アクティブ化されていることを示します。
- **39**：SGSN only：このコードは、GGSN の再起動後に、ネットワーク（GGSN）が PDP コンテキストの再アクティブ化を要求したことを示します。
- **40**：RADIUS 接続解除要求メッセージの受信により、GGSN が「delete PDP context request」を送信した場合。

**abnormal-release-cause**：このキーワードは、SGW-CDR の診断フィールドへのベアラーの異常終了情報の挿入を制御します。CLI コマンド「**gtpp attribute diagnostics**」は、**abnormal-release-cause** を無効にして **diagnostics** フィールドを有効にすることに注意してください。**no gtpp attribute diagnostics** コマンドは、**abnormal-release-cause** と **diagnostics** フィールドの両方を無効にします。

**重要** ベアラーの異常終了機能は現在、custom34 と custom35 の GTPP ディクショナリに対してのみ適用できます。つまり、レコード終了の原因が「Abnormal Release」の場合、custom34 ディクショナリと custom35 ディクショナリでは SGW-CDR に、custom35 GTPP ディクショナリでは PGW-CDR にベアラー終了の原因が挿入されます。

**direct-tunnel**

デフォルト：無効

PGW-CDR/eG-CDR の直接トンネルフィールドを挿入します。

このキーワードは、GGSN、P-GW、および S-GW にのみ適用できます。

**duration-ms**

必須の Duration フィールドに含まれる情報が秒単位ではなくミリ秒単位で報告されることを指定します（標準で求められている場合）。デフォルト：無効

**dynamic-flag**

デフォルト：有効

このキーワードは、x-CDR へのオプションフィールド「Dynamic Flag」の挿入を制御します。

**dynamic-flag-extension**

デフォルト：有効

このキーワードは、x-CDR へのオプションフィールド「Dynamic Address Flag Extension」の挿入を制御します。

このフィールドは、デュアル PDP コンテキストに IPv4 アドレスが動的に割り当てられている場合に、CDR に表示されます。この拡張機能フィールドは、デュアルスタックベアラーのサポートが使用可能になるように、3GPP リリース 10 に準拠した CDR で必要です。

**extended-bitrate**

デフォルト：無効

このキーワードは、APN-AMBR、MBR、または GBR が 4.2 Gbps を超える場合の P-GW CDR への拡張ビットレート情報の挿入を制御します。

**furnish-charging-information**

デフォルト：無効

このキーワードは、eG-CDR と PGW-CDR へのオプションフィールド「pSFurnishChargingInformation」の挿入を制御します。

重要 Furnish Charging Information（FCI）機能は、custom43 ディクショナリを除き、3GPP Rel.7 と 3GPP Rel.8 に準拠したすべての GTPP ディクショナリに適用されます。このキーワードオプションは、有効なライセンスがインストールされている場合にのみ使用できます。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。

GTPP サーバ グループ コンフィギュレーション モードで **gtpp attribute furnish-charging-information** コマンドを使用するなど、コマンドレベルで有効になっている場合にのみ、PGW-CDR と eG-CDR には FCI が含まれます。

FCI が変更されるたびに、新しい Free-Format-Data（FFD）の値は既存の FFD に付加されるか、または Append-Free-Format-Data（AFFD）フラグに応じて既存の FDD 上で上書きされます。CDR は FCI の変更時には生成されません。

FCI は、メイン CDR と LOSDV でもサポートされています。トリガー（ボリューム、時間、RAT など）が発生するたびに、コマンドレベルで現在使用可能な FFD が両方の CDR の本文に追加されます。コマンドレベルで次の Credit-Control-Answer メッセージによって付加または上書きされるまで、コマンドレベルの同じ FFD が次の CDR の本文に追加されます。

custom43 ディクショナリの場合、FCI の実装は次のようになります。

- FCI が変更されるたびに、PGW-CDR は CDR を生成します。つまり、古いバケットをクローズし、生成された CDR 内に古い FCI の詳細が保持されます。
- CDR 内の PS-Free-Format-Data の変換は、ASCII 形式の 16 進数値（0 ～ 9 の数値）から整数の 10 進値への変換です。
- PS-Append-Free-Format-Data は常に上書きします。

**imei**

デフォルト：無効

SGSN の場合：IMEI 値を S-CDR に挿入します。

GGSN の場合：IMEISV 値を G-CDR に挿入します。

**imsi-unauthenticated-flag**

デフォルト：有効

このキーワードは x-CDR へのオプションフィールド「IMSI Unauthenticated Flag」の挿入を制御します。

サービス対象の IMSI が認証されていない場合、このフィールド「IMSI Unauthenticated Flag」は、設定されていれば、custom35 ディクショナリの P-GW CDR レコードに存在します。このフィールドは、3GPP TS 32.298 v10.7 に従って追加されます。

**lapi**

デフォルト：無効

CDR に Low Access Priority Indicator（LAPI）フィールドを挿入します。このフィールドは、MTC 機能のサポートに必要です。

UE が優先順位の低い接続を示している場合は、「lowPriorityIndicator」属性が CDR に挿入されます。

**last-ms-timezone**

CDR フィールドに「Last MS-Timezone」を設定します。デフォルトのオプションが使用されている場合、このオプションは無効になります。

**last-uli**

CDR フィールドに「Last ULI」を設定します。デフォルトのオプションが使用されている場合、このオプションは無効になります。

**local-record-sequence-number**

デフォルト：無効

このキーワードは、ローカルレコードシーケンス番号とノードIDの両方を提供します。x-CDRでは、このフィールドはノードによって生成されたCDRの数が示されます。また、このフィールドはセッションマネージャ内で一意です。

PDP コンテキストがリリースされた場合や、設定に基づいて CDR の一部が生成された場合などのいくつかの理由で、Node ID フィールドが x-CDR に挿入されます。このフィールドは、SGSN サービスや GGSN サービスの名前に自動的に付加された AAA マネージャ識別子から構成されます。

Node ID フィールドの最大長は 20 バイトであるため、SGSN サービスまたは GGSN サービスの名前が切り捨てられることがあります。各 AAA マネージャは単独で CDR を生成するため、Local Record Sequence Number フィールドと Node ID フィールドは CDR を独自に識別できます。

重要 **gtpp single-source centralized-lrsn** が設定されている場合、「Node-ID」は指定した NodeID サフィックスのみで構成されます。NodeID サフィックスが設定されていない場合は、GTPP グループ名が使用されます。デフォルトの GTPP グループの場合、GTPP コンテキスト名が使用されます。**gtpp single-source centralized-lrsn** が設定されている場合は、Sessmgr によって生成された CDR のノード ID 形式は <1-byte Sessmgr restartvalue><3-byte Sessmgr instance number> <node-id-suffix> となります。**gtpp single-source centralized-lrsn** が設定されていない場合は、ACSmgr によって生成された CDR のノード ID 形式は <1-byte ACSmgr restart-value> <3-byte ACSmgr instance number> <Active charging service-name> となります。

**losdv**

デフォルト：有効

このキーワードは、x-CDRへのオプションフィールド「List of Service Data」の挿入を制御します。

**ms-timezone**

デフォルト：有効

このキーワードは、x-CDRへのオプションフィールド「MS-Timezone」の挿入を制御します。

**msisdn**

デフォルト：有効

このキーワードは、x-CDR へのオプションフィールド「MSISDN」の挿入を制御します。

**node-id**

デフォルト：有効

このキーワードは、x-CDR へのオプションフィールド「Node ID」の挿入を制御します。

**node-id-suffix 文字列**

デフォルト：無効

GTPP CDR の NodeID フィールドに使用するために設定した Node-ID-Suffix を 1 ～ 16 文字の英数値文字列として指定します。各セッションマネージャタスクは、GTPP コンテキストごとに一意の NodeID 文字列を生成します。

重要 NodeID フィールドは、*ndddSTRING* 形式の印刷可能な文字列です。*n*：最初の桁は、0 ～ 7 の値を持つ Sessmgr 再起動カウンタです。*add*：sessmgr インスタンスの数です。指定した NodeID-suffix をすべての CDR で使用します。「Node-ID」フィールドは、sessMgr リカバリカウンタ（1 桁）*n* + AAA マネージャ識別子（3 桁）*ddd* + 設定されたノード ID サフィックス（1 ~16 文字）の文字列で構成されます。一元化 LRSN 機能が有効になっている場合、「Node-ID」フィールドは、指定した NodeID-suffix のみで構成されます（NodeID-prefix は挿入されません）。このオプションが設定されていない場合は、代わりに GTPP グループ名が使用されます（デフォルトの GTPP グループの場合は context-name が使用されます）。

重要 この **node-id-suffix** が設定されていない場合、GGSN は GTPP コンテキスト名を Node-id-suffix を使用し（16 文字に切り捨てられます）、SGSN は GTPP グループ名を node-id-suffix として使用します。

**packet-count**

デフォルト：無効

このオプションを指定すると、CDR にオプションフィールド「datapacketFBCUplink」と「datapacketFBCDownlink」が挿入されます。

重要 このキーワードは、custom24 GTPP ディクショナリに適用されます。

**pco-nai**

このオプションを指定すると、P-GW CDR にオプションフィールド「PCO-Network Access Identifier」を挿入します。

重要 このキーワードは、custom44 GTPP ディクショナリに適用されます。

**pdn-connection-id**

デフォルト：有効

このキーワードは、x-CDR へのオプションフィールド「PDN Connection ID」の挿入を制御します。

**pdp-address**

デフォルト：有効

このキーワードは、x-CDR へのオプションフィールド「PDP Address」の挿入を制御します。

**pdp-type**

デフォルト：有効

このキーワードは、x-CDR へのオプションフィールド「PDP Type」の挿入を制御します。

**pgw-ipv6-addr**

デフォルト：無効

このオプションを指定すると、P-GW IPv6 アドレスを設定できます。

重要　この属性は、custom24 と custom35 の SGW-CDR ディクショナリでの設定を制御できます。

**pgw-plmn-id**

デフォルト：有効

このキーワードは、x-CDR へのオプションフィールド「PGW PLMN-ID」の挿入を制御します。

**plmn-id [ unknown-use ]**

デフォルト：有効

SGSN の場合は、ディクショナリがサポートしている場合に提供される S-CDR 内の SGSN PLMN 識別子の値（RAI）を報告します。

GGSN では、SGSN によって GTP の PDP コンテキスト作成要求でもともと提供されていた G-CDR 内で SGSN PLMN 識別子の値（RAI）を報告します。SGSN が提供していない場合は、省略されます。

通常、SGSN PLMN-id 情報が使用できない場合、属性 sgsnPLMNIdentifier は CDR に挿入されません。このキーワードを使用すると、SGSN PLMN-id が使用できない場合に、特定の値を持つ sgsnPLMNIdentifier を挿入できます。

**unknown-use** *hex_num*：PLMN-id を提供していない外部 SGSN を識別する 0x0 ～ 0xFFFFFF の 16 進数です（GGSN のみ）。

**qos max-length**

デフォルト：無効

このオプションを指定すると、S-CDR と SaMOG CDR 内で送信された QoS に関連するパラメータが変更されます。この **max-length** オプションは、CDR 内で送信される QoS の長さを変更するために使用されます。**qos_value** は 4 ～ 24 の整数である必要があります。

この機能は、Rel.7+ QoS の形式をサポートするために導入されました。

**rat**

デフォルト：有効

SGSN の場合：S-CDR 内の RAT（無線アクセステクノロジータイプを識別する）値を挿入します。

GGSN の場合：G-CDR 内の RAT（無線アクセステクノロジータイプを識別する）値が含まれます。

**recordextension**

デフォルト：無効

このキーワードは、x-CDR へのオプションフィールド「RecordExtension」の挿入を制御します。

**record-extensions rat**

デフォルト：無効

ネットワークオペレータやメーカーが、3GPP TS 32.298 リリース 7 以降の標準レコードの定義に従って、CDR に独自の推奨拡張を追加できるようにします。

**record-type { sgsnpdprecord | sgwrecord }**

重要　このキーワードは、SaMOG 混合モードライセンス（3G と 4G の両方をサポート）が設定されている場合にのみ使用できます。

デフォルト：sgwrecord

使用する SaMOG CDR タイプを指定します。

SaMOG 3G ライセンスの場合、このキーワードは使用できません。ただし、デフォルトのレコードタイプとして sgsnpdprecord タイプが使用されます。

**served-mnai**

デフォルト：無効

このキーワードは、x-CDR へのオプションフィールド「Served MNAI」の挿入を制御します。

**served-pdp-pdn-address-extension**

デフォルト：無効

IPv4v6 デュアルスタック PDP アドレスタイプのサポートでは、このキーワードを指定すると、サービスが IPv4v6 アドレス情報を CDR に挿入します。IPv4 アドレスが Served PDP PDN Address Extension フィールドに挿入され、IPv6 アドレスが Served PDP Address フィールドか Served PDP PDN Address フィールドに挿入されます。

重要　この属性は、GTPP ディクショナリが custom34 に設定されている場合は表示されません。

（注）　SGSN の場合、**served-pdp-pdn-address-extension** を有効にした時点で、次のディクショナリを除くすべてのカスタム S-CDR ディクショナリが CDR フィールド「Served PDP/ PDN Address extension」をサポートします。

- custom17
- custom18
- custom23
- custom42
- custom41

**served-pdp-pdn-address-prefix-length**

デフォルト：有効

IPv6 プレフィックス委任のサポートでは、このキーワードを指定すると、サービスが x-CDR にこの「Served PDP PDN Address」フィールドを挿入します。

このフィールドが設定されている場合、servedPDPPDNAddress フィールドは、3GPP 32.298 に概説されているように IPv6 プレフィックス長を報告できます。プレフィックス長は、次の場合にのみ報告されます。

- 設定されている
- デフォルト長の 64 ではない
- IPv6 コールか IPv4v6 コールである

**af-record-info**

デフォルト：無効

AF によって生成され、Gx インターフェイスを介して P-GW によって受信された**AF Charging Identifier**キーワードおよび関連するフロー識別子を含めるためのイネーブル属性。このキーワードは、custom24 GTPP ディクショナリに適用されます。

**sgsn-change**

デフォルト：有効

このキーワードは SGSN に固有であり、ライセンスが制限されています。

このキーワードは、S-CDR への S-CDR 属性「SGSN Change」の挿入を制御します。デフォルトで有効になっており、属性「SGSN Change」がデフォルトで S-CDR に挿入されます。

（注） SGSN 固有の custom33 ディクショナリの場合は、課金の問題を防ぐために、アップグレード前にこのキーワードを無効にすることを推奨します。

**sgw-ipv6-addr**

デフォルト：[Disabled]

このオプションを指定すると、S-GW IPv6 アドレスを設定できます。

重要 この属性は、custom24 と custom35 の SGW-CDR ディクショナリでの設定を制御できます。

**sms { destination-number | recording-entity | service-centre }**

このキーワードは SGSN に固有です。

このキーワードを入力すると、SMS-MO-CDR または SMS-MT-CDR に SMS 関連フィールドを挿入します。

**destination-number**：SMS-MO-CDR または SMS-MT-CDR に「destinationNumber」フィールドを挿入します。

**recording-entity**：SMS-MO-CDR または SMS-MT-CDR に「recordingEntity」フィールドを挿入します。

**service-centre**：SMS-MO-CDR または SMS-MT-CDR に「serviceCentre」フィールドを挿入します。

**sna-ipv6-addr**

デフォルト：[Disabled]

このオプションを指定すると、サービスノードの IPv6 アドレス（SNAv6）を設定できます。

重要 この属性は、custom24 と custom35 の SGW-CDR ディクショナリでの設定を制御できます。

**sponsor-id**

デフォルト：[Disabled]

PGW-CDR に Sponsor ID フィールドと Application-Service-Provider-Identity フィールドを挿入します。

PCEF がスポンサー提供データの接続機能をサポートしている場合は PGW-CDR に「Sponsor ID」属性と「Application-Service-Provider-Identity」属性が挿入されます。または、3GPP TS 29.212 で説明したとおり、必要なレポートレベルはスポンサー提供の接続になります。

この機能は、CDR についてのリリース 11 3GPP 仕様に準拠するように実装されます。したがって、リリース 11 に準拠する（custom35）すべての GTPP ディクショナリにこの動作が適用されます。

**start-time**

デフォルト：有効

このキーワードは、x-CDR へのオプションフィールドの「Start-Time」の挿入を制御します。

**stop-time**

デフォルト：有効

このキーワードは、x-CDR へのオプションフィールドの「Stop-Time」の挿入を制御します。

**twanuli**

デフォルト：[Disabled]

このキーワードは、CDR へのオプションフィールドの「TWAN User Location Information」の挿入を制御します。

**ue-tun-ip-port**

デフォルト：無効

21.9.5 以降のリリースでは、S2b（VoWifi）コール/サブスクライバに生成された CDR に新しいパラメータを挿入するようにこのキーワードが P-GW に導入されています。

重要　このキーワードは、custom24 GTPP ディクショナリに適用されます。

**uwanuli**

デフォルト：無効

このキーワードは、CDR へのオプションフィールド「UWAN User Location Information」の挿入を制御します。

**uli**

デフォルト：有効

このキーワードは、CDR にオプションフィールド「ユーザロケーション情報」を含めることを制御します。

**user-csg-information**

デフォルト：[Disabled]

このキーワードは、x-CDR へのオプションフィールド「User CSG Information」の挿入を制御します。

**重要** 現時点では、UCI 値は SGW-CDR でのみサポートされています。

この属性は、GTPP ディクショナリが custom11、custom34、または custom35 に設定されている場合は表示されません。

**+**

複数の属性を設定するために、このコマンドを複数回入力できることを示します。

**使用上のガイドライン** このコマンドは、GGSN によって生成された CDR 内で報告する必要があるオプションの情報フィールドの一部を決定します。さらに、一部の必須フィールドの情報がどのように報告されるかを制御します。

標準ではオプションとして記載されており、上記に記載されていないフィールドは、レコードの拡張機能の場合を除き（存在することはない）、CDR に常に存在するようになります。

**例**

次のコマンドは、S-CDR への「SGSN Change」フィールドの挿入を無効にします。

```
no gtpp attribute sgsn-change
```

**例**

次のコマンドは、CDR 内の Duration フィールドに指定された時間をミリ秒単位で報告することを決定します。

```
gtpp attribute duration-ms
```

# gtpp trigger

部分的な CDR レコードのクローズまたは新しい CDR レコードコンテナのオープンのいずれかの原因となる GTPP トリガー条件を無効にします。GTPP トリガーは 3GPP TS 32.251 v6.6.0 で指定されています。GTPP トリガーのすべての 変更は、**volume-limit** を除き、ただちに有効になります。

**Command_Product** ECS

GGSN

P-GW

SAEGW

SGSN

S-GW

**Command_Privilege**

セキュリティ管理者、管理者

**コマンド モード**

Exec > グローバル コンフィギュレーション > コンテキスト コンフィギュレーション > GTPP サーバ グループ コンフィギュレーション

**configure > context** *context_name* **> gtpp group** *group_name*

上記のコマンドシーケンスを入力すると、次のプロンプトが表示されます。

```
[context_name]host_name(config-gtpp-group)#
```

**構文の説明**

```
gtpp trigger { apn-ambr-change | [ default-bearer-only |
all-non-gbr-bearers | all-bearers ] | cell-update |
ciot-userplane-change| dcca | direct-tunnel | egcdr max-losdv |
ggsn-preservation-mode-change | inter-plmn-sgsn-change |
ms-timezone-change | plmn-id-change | qos-change | rat-change [ generate
 { cdr | container } ] | routing-area-update | service-idle-out |
serving-node-change-limit | sgsn-change-limit | tariff-time-change |
time-limit | uli-change | volume-limit }
default gtpp trigger
no gtpp trigger { apn-ambr-change | [ default-bearer-only |
all-non-gbr-bearers | all-bearers ] | cell-update |
ciot-userplane-change| dcca | direct-tunnel | egcdr max-losdv |
ggsn-preservation-mode-change | inter-plmn-sgsn-change |
ms-timezone-change | plmn-id-change | qos-change | rat-change [ generate
 { cdr | container } ] | routing-area-update | service-idle-out |
serving-node-change-limit | sgsn-change-limit | tariff-time-change |
time-limit | uli-change | volume-limit }
```

**default**

指定されたトリガー条件をデフォルト設定に戻します。デフォルトでは、すべてのトリガー条件が有効になっています。

**no**

指定されたトリガー条件を無効にします。

**apn-ambr-change [ default-bearer-only | all-non-gbr-bearers | all-bearers ]**

デフォルト：無効

デフォルトのベアラー、あるいはその PDN のすべてのベアラーまたは選択した apn-non-gbr ベアラーに対してのみ APN AMBR トリガーを有効にします。

重要 このキーワードオプションは、有効なライセンスがインストールされている場合にのみ使用できます。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。

APN集約最大ビットレート（AMBR）は、APNごとに保存されるサブスクリプションパラメータです。これにより、すべてのGBR以外のベアラーと同じAPNのすべてのPDN接続にわたって提供されることが予想される集約ビットレートが制限されます。これらのGBR以外のベアラーのそれぞれがAPN AMBR全体を利用する可能性があります。たとえば、他のGBR以外のベアラーがトラフィックを伝送しない場合などです。

15.0以降のリリースでは、このCLIコマンドを次の追加オプションとともに設定して、すべてのGTPPディクショナリ内のSGW-CDRに対するAPN-AMBRトリガーを有効にする必要があります。

- **default-bearer-only**：コンテナをデフォルトのベアラーのみに追加します。
- **all-non-gbr-bearers**：コンテナをすべての非gbrベアラーに追加します。
- **all-bearers**：すべてのベアラーにコンテナを追加します。

重要 このCLIコマンドおよび関連付けられているオプションは、S-GWとP-GW以外の製品では使用できません。

各CDRの最初のコンテナには、QoSとともにapn-ambrフィールドが含まれています。次のコンテナでは、以前の変更条件が「QoSの変更」または「APN AMBRの変更」の場合、このフィールドが表示されます。

**cell-update**

**gtpp dictionary** 設定内に指定されているディクショナリにセル更新のサポートが含まれている場合は、S-CDRのセル更新トリガーを有効にします。このトリガーは、2Gでのみ使用できます。現時点では、custom18ディクショナリがセル更新トリガーをサポートしています。

**ciot-userplane-change**

CDRのユーザプレーンの変更トリガーを有効にします。

**dcca**

このキーワードは、次のDCCAによって生成されたトリガーに対して、PGW-CDRでのLOSDVの追加を有効または無効にします。

- 時刻のしきい値に到達した
- ボリュームのしきい値に到達した
- サービス固有ユニットのしきい値に到達した
- 時間を使い果たした
- ボリュームを使い果たした
- 有効性タイムアウト

- 再許可要求
- 進行中のセッションの続行
- 進行中のセッションの再試行と終了
- 進行中のセッションの終了
- サービス固有のユニットが枯渇した
- エンベロープのクローズ

**direct-tunnel**

CDR の直接トンネルトリガーを有効にします。

**egcdr max-losdv**

サービスデータボリュームのリスト（LoSDV）コンテナが LOSDV コンテナに設定されている制限を超えた場合に eG-CDR/P-CDR のトリガーを有効にします。デフォルト：無効

**ggsn-preservation-mode-change**

このキーワードは GGSN のみを対象としています。

このトリガーにより、G-CDR の preservation-mode-changeトリガーが有効になります。

**inter-plmn-sgsn-change**

このキーワードは GGSN のみを対象としています。

このトリガーを無効にすると、PLMN 間 SGSN の変更は無視され、G-CDR はリリースされません。デフォルト：有効

**ms-timezone-change**

このキーワードは GGSN に固有です。

このトリガーが無効になっている場合、タイムゾーンの変更に対するレコードの部分的なクローズは発生しません。MS タイムゾーンの変更は、3GPP R6 ベースの GTPP ディクショナリにのみ適用する必要があります。デフォルト：有効

**plmn-id-change**

このトリガーキーワードは、2G SGSN に固有であり、独自の（非標準）ものです。

**gtpp dictionary** 設定で指定されたディクショナリが PLMNID の変更をサポートしている場合は、S-CDR の PLMNID 変更トリガーを有効にします。有効にすると、SGSN の下にある間に MS が PLMN を変更したときに（システム内の SGSN 内での PLMN-ID のハンドオーバー）SGSN は S-CDR を部分的に生成します。現時点では、custom18 ディクショナリがこのトリガーをサポートしています。デフォルト：無効

**qos-change**

CDR の QoS 変更トリガーを有効にします。このトリガーを無効にすると QoS の変更は無視され、新しい CDR は開かれません。デフォルト：有効

QoS の変更が確認されると、システムはコンテナのみを生成します。ただし、最大コンテナの条件に到達すると、暫定 CDR が生成されます。

**rat-change [ generate { cdr | container } ]**

RAT の変更の部分的なレコードクローズを有効または無効にします。無効にした場合は、RAT の変更に対する部分的なレコードクローズは発生しません。RAT の変更は、3GPP R6 ベースの GTPP ディクショナリのみに適用する必要があります。デフォルト：有効

SGSN では、RAT の変更トリガー（2G <-> 3G）は、サービス間ハンドオフ（SGSN サービス <-> GPRS サービス）を意味します。このトリガーが有効になっている場合は、RAT の変更後に暫定 CDR が生成されます。この RAT が CDR を変更すると、ボリューム/時刻などの CDR のしきい値や GTPP グループが新しいサービスから適用されます。RAT の変更トリガーが無効になっている場合は、CDR しきい値と GTPP グループなどは変更されず、古いサービスのものが引き続き使用されます。

RAT が変更されると、CDR の [System Type] フィールドが変更され、新しいシステムタイプが表示されます。このトリガーが無効になっている場合は、生成された次の CDR がシステムタイプを示しますが、CDR 内のデータカウントは CDR に示されているシステムタイプに帰属しているとは限りません。代わりに、ハンドオーバーの実行時に CDR が閉じていないため、2G と 3G の両方に帰属している場合があります。

重要 CDR 関連の変更の [System Type] フィールドは、[System Type] フィールドを使用しないカスタマイズされた CDR フォーマットには適用されません。

**generate { cdr | container }**：RAT 変更時の CDR の生成か、またはコンテナの生成のみを設定します。

**cdr**：RAT 変更時に CDR を生成します。

**container**：RAT 変更時のみにコンテナを生成します。

**routing-area-update**

CDR のルーティングエリア更新トリガーを有効にします。

**service-idle-out**

このキーワードは、サービスがアイドルアウトした場合の PGW-CDR 内での LOSDV の追加を有効または無効にします。

CDR モジュールは、クォータホールドタイマーが期限切れになったときに DCCA モジュールから、またはルールベースにサービスのアイドルアウト設定がある場合は ACS から、サービスのアイドルアウトトリガーを受信します。

**serving-node-change-limit [ also-intra-sgsn-multiple-address-group-change ]**

このキーワードは、P-GW、S-GW、および GGSN に対して有効になります。ただし、**also-intra-sgsn-multiple-address-group-change** は、GGSN に対してのみ有効になります。デフォルト：有効

このトリガーを無効にすると、SGSN の変更は無視され、SGSN の IP アドレスは CDR の SGSN アドレスリストに追加されません。これにより、設定された制限を超えた SGSN の変更によって、CDR のリリースを低減できます。

**also-intra-sgsn-multiple-address-group-change**：このキーワードには、SGSN 変更としての SGSN グループ内の変更が含まれています。

**sgsn-change-limit [ also-intra-sgsn-multiple-address-group-change ]**

このキーワードは廃止されていますが、既存の顧客の下位互換性を維持するために使用できます。**sgsn-change-limit** の新しいキーワードは、**serving-node-change-limit** です。デフォルト：有効

このトリガーを無効にすると、SGSN の変更は無視され、SGSN の IP アドレスは CDR の SGSN アドレスリストに追加されません。これにより、設定された制限を超えた SGSN の変更によって、CDR のリリースを低減できます。

**also-intra-sgsn-multiple-address-group-change**：このキーワードには、SGSN 変更としての SGSN グループ内の変更が含まれています。

**tariff-time-change**

このトリガーを無効にすると、タリフ時間の変更に対してコンテナのクローズは実行されません。デフォルト：有効

このトリガーは、MBMS セッションの G-MB-CDR にも適用されます。

**time-limit**

このトリガーが無効になっている場合は、設定された時間制限に到達しても、レコードの部分的なクローズは実行されません。デフォルト：有効

このトリガーは、MBMS セッションの G-MB-CDR にも適用されます。

**uli-change**

GTPP ディクショナリ設定に指定されているディクショナリにユーザロケーション更新トリガーのサポートが含まれている場合に、eG-CDR/PGW-CDR/SGW-CDR のユーザロケーション更新トリガーを有効にします。デフォルト：有効

**volume-limit**

このトリガーが無効になっている場合は、ボリューム制限に到達しても、レコードの部分的なクローズは実行されません。デフォルト：有効

このトリガーは、MBMS セッションの G-MB-CDR にも適用されます。

**使用上のガイドライン**

このコマンドを使用して、CDR レコードの部分的なクローズか、または新しい CDR の作成を引き起こす可能性がある GTTP トリガーを無効または有効にします。

**例**

次に、設定された時間制限に到達したときにレコードの部分的なクローズを無効にするコマンドを示します。

```
gtpp trigger time-limit
```

次に、設定された時間制限に到達したときにレコードの部分的なクローズを再度有効にするコマンドを示します。

```
no gtpp trigger time-limit
```

# gtpu-error-ind

RNC、SGSN、または P-GW から GTP-U エラーインジケータを受信したときに実行されるアクションを設定します。

**Command_Product**

S-GW

SAEGW

**Command_Privilege**

管理者

**コマンド モード**

[Exec] > [Global Configuration] > [Context Configuration] > [S-GW Service Configuration]

**configure > context** *context_name* **> sgw-service** *service_name*

上記のコマンドシーケンスを入力すると、次のプロンプトが表示されます。

```
[context_name]host_name(config-sgw-service)#
```

**構文の説明**

```
gtpu-error-ind { { s12 | s1u | s11u } { local-purge | page-ue [
custom1-behavior ] } | { s4u | s5u } { local-purge | signal-peer } }
default gtpu-error-ind { s12 | s1u | s11u | s4u | s5u }
```

**default**

指定されたインターフェイスのデフォルトアクションにコマンドをリセットします。S12 および S1-U の場合 **page-ue** は、デフォルトのアクションです。S4-U および S5-U の場合 **local-purge** は、デフォルトのアクションです。

**{ s12 | s1u | s11u }{ local-purge | page-ue [ custom1-behavior ] }**

S12 インターフェイスを介して、または S1-U インターフェイスを介して、無線ネットワークコントローラ（RNC）から GTP-U エラー通知を受信した場合に実行するアクションを指定します。

**local-purge**：S-GW は、ピアに通知せずに、影響を受けるベアラー（またはデフォルトのベアラーでエラー通知が受信された場合は PDN）をクリアします。

**page-ue [ custom1-behavior ]**：S-GW は、完全な状態を S1-Idle に移行し、UE のページングを開始します。カスタム 1 動作オプションが指定されている場合は、S-GW は 60 秒のタイマーでページングの試行をガードします。この時間内に、ベアラーは MME によって eNodeB TEID をリフレッシュする必要があります。それ以外の場合、S-GW は影響を受けるベアラーをシグナリングでクリアします。これは、S12 および S1-U インターフェイスで受信した GTP-U エラー通知メッセージのデフォルトアクションです。

**{ s4u | s5u }{ local-purge | signal-peer }**

S4 インターフェイスを介して SGSN から、または S5-U インターフェイス経由で P-GW から GTP-U エラー通知を受信した場合に実行するアクションを指定します。

**local-purge**：S-GW は、ピアに通知せずに、影響を受けるベアラー（またはデフォルトのベアラーでエラー通知が受信された場合は PDN）をクリアします。これは、S4 および S5-U インターフェイスで受信した GTP-U エラー通知メッセージのデフォルトアクションです。

**signal-peer**：S-GW は、ピア MME と P-GW に向けた制御シグナリングを開始します。シグナリングの場合：

- ベアラーの削除の場合、S-GW は P-GW に Delete-Bearer-Command メッセージを、MME に Delete-Bearer-Request（EBI を使用）メッセージを送信します。
- PDN の削除の場合、S-GW は P-GW に Delete-Session-Request メッセージを、MME に Delete-Bearer-Request（LBI を使用）メッセージを送信します。
- S-GW は、ピアからの削除応答を待機しません。要求は一度のみ送信され、ローカルリソースがリセットされます。

**使用上のガイドライン** このコマンドを使用して、S12 インターフェイスを介して RNC から受信した GTP-U エラー表示、S1-U インターフェイスを介して eNodeB から受信した GTP-U エラー表示、S4-U インターフェイスを介して SGSN から受信された GTP-U エラー表示、または S5-U インターフェイスを介して P-GW から受信された GTP-U エラー表示を受信したときの動作を指定します。

**例**

次のコマンドは、eNodeB から GTP-U エラー通知を受信したときに、影響を受けるベアラーをクリアするために実行するアクションを設定します。

```
gtpu-error-ind s1u local-purge
```

# ie-override

このコマンドは、MME から HSS に送信されるメッセージに対して設定された値を使用して、RAT タイプの AVP 値を上書きするために使用されます。

**重要** このコマンドは、HSS が新しい NB-IoT RAT タイプをサポートしていないため、以前のリリースとの下位互換性を保証します。

**Command_Product**

MME

**Command_Privilege**

管理者

**コマンド モード**

[Exec] > [Global Configuration] > [Call Control Profile Configuration]

**configure > call-control-profile** *profile_name*

上記のコマンドシーケンスを入力すると、次のプロンプトが表示されます。

`[local]host_name(config-call-control-profile-profile_name)#`

**構文の説明**

**[ remove ] ie-override s6a rat-type wb-eutran**

**remove**

キーワードの remove を指定すると、既存の設定が削除されます。

**ie-override**

このキーワードを使用すると、MME から HSS に送信されるメッセージで、オペレータが IE オーバーライドを設定できます。

**s6a**

このキーワードは、インターフェイスを s6a として指定するために使用されます。ホームサブスクライバサーバ（HSS）と通信するために MME によって使用される s6a インターフェイス。

**rat-type**

このキーワードを使用すると、サポートされている RAT タイプの AVP IE を設定できます。

**wb-eutran**

このキーワードを使用すると、WB-EUTRAN AVP 値を指定できます。

**使用上のガイドライン**

このコマンドを使用すると、s6a インターフェイスを介して MME から HSS に送信されるメッセージに設定された値を使用し、RAT タイプの AVP 値を上書きできます。設定された RAT タイプが NB-IoT の場合、MME から HSS に送信されるメッセージのために wb-eutran に変更されます。このコマンドはデフォルトでは無効になっています。

**例**

次のコマンドは、設定された値 WB EUTRAN を使用して RAT タイプ AVP 値のオーバーライドを有効にするために使用されます。

```
ie-override s6a rat-type wb-eutran
```

# iftask mcdmatxbatch

マルチチャネル ダイレクト メモリ アクセス（MCDMA）の送信バッチ処理を設定します。MCDMA は、IFTASK から SESSMGR へのパスです。このコマンドは、仮想化プラットフォームの StarOS にのみ適用されます。

**Command_Product**

すべて

**Command_Privilege**

演算子

**コマンド モード**

[Exec] > [Global Configuration]

**configure**

上記のコマンドシーケンスを入力すると、次のプロンプトが表示されます。

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

**構文の説明**

**[ no ] iftask mcdmatxbatch { burstsize** *number_of_packets* **| latency** *milliseconds* **}**

**no**

Iftask mcdmatxbatch の設定を削除します。

**burstsize *number_of_packets***

1 ～ 1024 のバーストあたりの最大パケット数。

**latency *milliseconds***

現在、サポートされていません。

**使用上のガイドライン**

次に、MCDMA のバーストあたりの最大パケット数を 512 に設定する例を示します。

```
iftask mcdmatxbatch burstsize 512
```

# iftask txbatch

送信バッチ処理を設定します。このコマンドは、仮想化プラットフォームの StarOS にのみ適用されます。

**Command_Product**

すべて

**Command_Privilege**

演算子

**コマンド モード**

[Exec] > [Global Configuration]

**configure**

上記のコマンドシーケンスを入力すると、次のプロンプトが表示されます。

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

**構文の説明**

**[ no ] iftask txbatch { burstsize** *number_of_packets* **| flush_latency | latency** *milliseconds* **}**

**no**

iftask txbatch の設定を削除します。

**burstsize *number_of_packets***

イーサネット インターフェイスに送信する前にベクターに蓄積するパケットの最大数を 1 ～ 1024 の間で指定します。

**latency *milliseconds***

現在、サポートされていません。

**使用上のガイドライン**

このコマンドを使用して、システム全体の IFTASK 操作の送信バッチ処理パラメータを設定します。

次に、MCDMA のバーストあたりの最大パケット数を *512* に設定する例を示します。

```
iftask txbatch burstsize 512
```

次の例では、制御ポートのバイトをフラッシュするための最大待機時間を *1000* ミリ秒に設定しています。

```
iftask txbatch flush_latency 1000
```

# ip name-servers

現在のコンテキストが論理ホストの名前解決に使用する可能性があるドメインネームサーバのリストを変更します。

**Command_Product** すべて

**Command_Privilege** セキュリティ管理者、管理者

**コマンド モード** [Exec] > [Global Configuration] > [Context Configuration]

**configure > context** *context_name*

上記のコマンドシーケンスを入力すると、次のプロンプトが表示されます。

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

**構文の説明**

**ip name-servers** *ip_address secondary_ip_address* [*third_ip_address*]
**no ip name-servers** *ip_address*

**no**

指定したネームサーバを現在のコンテキストのネームサーバのリストから削除することを示します。

***ip_address***

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、ドメインネームサーバの IP アドレスを指定します。

***secondary_ip_address***

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 進表記のいずれかを使用して、セカンダリ ドメイン ネーム サーバの IP アドレスを指定します。

***third_ip_address***

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 進表記のいずれかを使用して、第 3 のドメインネームサーバの IP アドレスを指定します。（VPC のみ）

**使用上のガイドライン** 現在のコンテキストで論理ホスト名の解決に使用される可能性があるネームサーバのリストを管理します。

DNS は、コンテキスト コンフィギュレーションのコンテキストレベルと、APN コンフィギュレーション モードの APN レベルで **dns** コマンドと **ipv6 dns** コマンドを使用して指定できます。また、AAA サーバから受け取ることもできます。

PCO 設定で DNS が要求されると、DNS 値は次の優先順位に従います。

1. LNS から受信した DNS 値が最優先される。
2. RADIUS サーバから受信した DNS 値の優先順位は 2 番目になる。
3. **dns** コマンドと **ipv6 dns** コマンドを使用して APN でローカルに設定した DNS 値の優先順位は 3 番目になる。
4. コンテキストレベルで設定した DNS 値の優先順位は最下位になる。

重要　同じ優先順位を NBNS サーバに適用し、ICPC を介して LNS とネゴシエートすることができます。

**例**

```
ip name-servers 10.2.3.4
```

# ip qos-dscp

この APN プロファイルの IP パラメータを定義します。

**Command_Product**

MME

SGSN

S-GW

SAEGW

**Command_Privilege**

管理者

**コマンド モード**

[Exec] > [Global Configuration] > [APN Profile Configuration]

**configure > apn-profile** *profile_name*

上記のコマンドシーケンスを入力すると、次のプロンプトが表示されます。

`[local]host_name(apn-profile-profile_name)#`

**構文の説明**

**ip { qos-dscp { { downlink | uplink } { background***forwarding* **| conversational** *forwarding* **| interactive traffic-handling-priority** *priority forwarding* **| streaming** *forwarding* **} + }  | source-violation { deactivate [ all-pdp | exclude-from accounting | linked-pdp | tolerance-limit } | discard [ exclude-from-accounting ] | ignore }**
**default ip { qos-dscp [ downlink | uplink ] | source-violation }**
**no ip qos-dscp { downlink | uplink } { background | conversational | interactive | streaming } +**

重要　具体的に設定されていないすべてのパラメータは、デフォルト値を使用して設定に含められます。

**default**

設定をデフォルト値にリセットします。

**no**

指定された IP QoS-DSCP マッピングを無効にします。

**qos-dscp**

特定の 3GPP QoS クラスのパケットを送信するために使用される DiffServe コードポイント（DCSP）マーキングを設定します。

**downlink | uplink**

ダウンリンク（ネットワークからサブスクライバへ）またはアップリンク（サブスクライバからネットワークへ）のいずれかの方向のパケットを設定します。**downlink** と **uplink** の設定には、次の 1 つ以上を含める必要があります。

- **background** ：3GPP バックグラウンドクラスに登録されたセッションのパケットに使用する DSCP マーキングを設定します。DSCP マーキングが続く必要があります。
- **conversational** ：3GPP 会話型クラスに登録されたセッションのパケットに使用する DSCP マーキングを設定します。DSCP マーキングが続く必要があります。
- **interactive** ：3GPP 双方向型クラス内の異なるトラフィック優先順位に登録されているセッションのパケットに使用する DSCP マーキングを設定します。トラフィック処理の優先順位（THP）1、2、または 3 が続く必要があります、
- **streaming** ：3GPP ストリーミングクラスに登録されたセッションのパケットに使用する DSCP マーキングを設定します。DSCP マーキングが続く必要があります。

### *DSCP* マーキングのオプション

ダウンリンクとアップリンクには、DSCP 転送マーキングが含まれている必要があります。サポートされているオプションは次のとおりです。

- af11：相対的優先転送 11 PHB の使用を指定します。
- af12：相対的優先転送 12 PHB の使用を指定します。
- af13：相対的優先転送 13 PHB の使用を指定します。
- af21：相対的優先転送 21 PHB の使用を指定します。
- af22：相対的優先転送 22 PHB の使用を指定します。
- af23：相対的優先転送 23 PHB の使用を指定します。
- af31：相対的優先転送 31 PHB の使用を指定します。
- af32：相対的優先転送 32 PHB の使用を指定します。
- af33：相対的優先転送 33 PHB の使用を指定します。
- af41：相対的優先転送 41 PHB の使用を指定します。
- af42：相対的優先転送 42 PHB の使用を指定します。
- af43：相対的優先転送 43 PHB の使用を指定します。
- be：ベストエフォートの転送 PHB の使用を指定します
- ef：完全優先転送 PHB の使用を指定します。

アップリンクとダウンリンクの両方のデフォルトの転送は次のとおりです。

- 会話型：ef;
- ストリーミング：af11;
- 対話型1：ef;
- 対話型 2：af21;
- 対話型 3：af21;
- バックグラウンド：be

**source-violation**

次の条件のいずれかを使用して、IP 送信元の違反の検出に関連する設定を行います。

- **deactivate** ：次の条件のいずれかを使用して PDP コンテキストを非アクティブにします。
  - **all-pdp** ：MS/UE-V のすべての PDP コンテキストを非アクティブにします。デフォルトでは、誤りのある PDP コンテキストを非アクティブにします。
  - **exclude-from-accounting** ：アカウンティングレコードで使用されている統計情報から無効な送信元 IP アドレスを持つパケットを除外します。
  - **linked-pdp** ：関連付けられているすべての PDP コンテキスト（プライマリとセカンダリ）を非アクティブにします。デフォルトでは、誤りのある PDP コンテキストを非アクティブにします。
  - **tolerance-limit** ：セッションを非アクティブにする前に許可する IP 送信元の最大違反数を設定します。
- **discard** ：誤りのあるパケットを破棄します。また、次のオプションを含めることができます。
  - **exclude-from-accounting** ：アカウンティングレコードで使用されている統計情報から無効な送信元 IP アドレスを持つパケットを除外します。
- **ignore** ：MS/UE IP 送信元の違反についてのパケットの確認を無視します。

**使用上のガイドライン** このコマンドは、APNプロファイルに関連付けられるIP機能の範囲を設定します。たとえば、次のようになります。

- 検出された IP 送信元の違反に対する応答での SGSN/S GW アクション
- トラフィッククラスごとのダウンリンクとアップリンクの設定の DSCP マーキング
- QoS クラス DiffServ コード

### 例

次に、APN プロファイルを設定して SGSN または S-GW に IP 送信元の違反情報を着信パケットで確認しないように指示するコマンドを示します。

```
ip source-violation ignore
```

# nb-iot

このコマンドは、Extended Discontinuous Reception（eDRX）を有効にし、MME 上で NB-IoT サブスクライバのそれぞれのパラメータを設定します。

**Command_Product**

MME

**Command_Privilege**

管理者

**コマンド モード**

[Exec] > [Global Configuration] > [Call Control Profile Configuration]

**configure > call-control-profile** *profile_name*

上記のコマンドシーケンスを入力すると、次のプロンプトが表示されます。

```
[local]host_name(config-call-control-profile-profile_name)#
```

**構文の説明**

```
nb-iot  {  edrx { ptw ptw_value edrx-cycle cycle_length_value | ue-requested
 } [ dl-buf-duration [ packet-count packet_count_value ] ] |
mo-exception-data reporting-threshold-value threshold_value  }
remove nb-iot {  edrx  | mo-exception-data }
```

#### remove

このキーワードは、NB-IoT サブスクライバの MME での eDRX 設定を無効にします。

#### edrx

このキーワードは、拡張された連続しない受信パラメータを設定します。

#### ptw *ptw_value*

このキーワードは、ページング時間枠（PTW）の値を設定します。*ptw_value* は、秒単位の整数値である必要があります。使用可能な値は、2.56、5.12、7.68、10.24、12.80、15.36、17.92、20.48、23.04、25.60、28.16、30.72、33.28、35.84、38.40、および 40.96 秒です。

#### ue-requested

このキーワードは、受け入れる接続要求メッセージまたは TAU 要求メッセージ内の UE から受信した Paging Time Window（PTW）と eDRX サイクル長の UE 要求値を指定します。

**edrx-cycle *cycle_length_value***

このキーワードは、eDRX サイクルの長さを設定します。*cycle_length_value* は、秒単位の整数値です。使用可能な値は、5.12、7.68、10.24、12.80、15.36、17.92、20.48、40.96、81.92、163.84、327.68、655.36、1310.72、2621.44、5242.88、および 10485.76 秒です。

**dl-buf-duration**

このオプションのキーワードは、UE をページングできない場合に DDN ACK 内でダウンリンクバッファ期間を送信します。

**packet-count *packet_count_value***

このオプションのキーワードは、UE をページングできない場合に、DDN ACK 内で「DL Buffering Suggested Packet Count」を送信します。*packet_count_value* は、0 ～ 65535 の整数値です。*packet_count_value* がローカルに設定されていない場合、*packet_count_value* のサブスクリプションに指定された値が使用されます。サブスクリプション値を 0 にすることができます。この場合、ローカルで設定されている場合でも、そのサブスクライバに対してパケットカウント IE は送信されません。

**mo-exception-data**

NBIOT RRC が MO の例外データカウンタを設定します。

**reporting-threshold-value*value***

レポートしきい値を指定します。*value* 1 ～ 50 の整数である必要があります。

## 使用上のガイドライン

このコマンドを使用して、MME 上で NB-IoT サブスクライバの eDRX を有効にします。オペレータは、次のコマンドを使用できます。

- eDRX パラメータの受け入れ：Paging Time Window（PTW）および eDRX cycle length value（UE-V）
- PTW および eDRX サイクルの長さの値を設定します。
- DDN ACK でのダウンリンクバッファ期間の設定（UE をページングできない場合）
- DDN ACK で「DL バッファリングの推奨パケット数」を設定します。

eDRX 機能が MME で有効になっている場合、有効なページングの場合にのみ、NB-IoT サブスクライバがページングされます。MME は、ページング中に、NB-IoT eDRX ページングパラメータを eNodeB に送信します。オペレータは、UE が要求した値を受け入れるようにオプションを設定するか、このコマンドを使用して値を設定することができます。このコマンドはデフォルトでは無効になっています。

同様の CLI コマンドが WB-EUTRAN サブスクライバに実装されています。WB-UTRAN eDRX パラメータおよび NB-IoT eDRX パラメータの両方を、WB-UTRAN および NB-IoT のサブスクライバのシステムに設定できます。

詳細については、『*Mme Administration Guide*』の「*eDRX Support on the MME*」機能の章を参照してください。

**例**

次のコマンドは、PTW と eDRX サイクル長を設定します。また、このコマンドは、推奨されるパケット数とともに、DDN ACK 内でダウンロードバッファ期間を送信します。

```
nb-iot edrx ptw 256 edrx-cycle 512 dl-buf-duration packet-count 10
```

# path-failure

S-GW と MME、P-GW、RNC、SGSN、または eNodeB の間でパス障害が発生したときに実行するアクションを設定します。

**Command_Product**

S-GW

SAEGW

**Command_Privilege**

管理者

**コマンド モード**

[Exec] > [Global Configuration] > [Context Configuration] > [S-GW Service Configuration]

**configure > context** *context_name* **> sgw-service** *service_name*

上記のコマンドシーケンスを入力すると、次のプロンプトが表示されます。

`[context_name]host_name(config-sgw-service)#`

**構文の説明**

```
path-failure { s11 | s11u | s12 | s1u | s4 | s4u | s5 | s5u } (
local-purge | signal-peer }
default path-failure { s11 | s11u | s12 | s1u | s4 | s4u | s5 | s5u }
( local-purge | signal-peer }
```

**default**

選択したインターフェイスのデフォルト設定である「local purge」にコマンドを返します。

**{ s11 | s12 | s1u | s4 | s4u | s5 | s5u }**

アクションを適用するインターフェイスを指定します。

**s11**：パス障害に対するアクションを、S-GW と MME の間の S11 インターフェイスに適用します。

**s11u**：パス障害に対するアクションを、S-GW と MME の間の S11-U インターフェイスに適用します。

**s12**：パス障害に対するアクションを、S-GW と RNC の間の S12 インターフェイスに適用します。

**s1u**：パス障害に対するアクションを、S-GW と eNodeB の間の S1-U インターフェイスに適用します。

**s4**：パス障害に対するアクションを、S-GWと SGSNの間の S4 制御プレーンインターフェイスに適用します。

**s4u**：パス障害に対するアクションを、S-GWと SGSNの間の S4-U ユーザ プレーン インターフェイスに適用します。

**s5**：パス障害に対するアクションを、S-GW と P-GW の間の S5 インターフェイスに適用します。

**s5u**：パス障害に対するアクションを、S-GW と P-GW の間の S5U ユーザ プレーン インターフェイスに適用します。

**{ local-purge | signal-peer }**

選択したインターフェイスに適用するアクションを指定します。

**local-purge**：S-GW は、ピアに通知せずに、影響を受けるベアラー（またはデフォルトのベアラーでパスの障害を受信した場合はPDN）をクリアします。これは、すべてのインターフェイスに対するデフォルトのアクションです。

**signal-peer**：S-GW は、ピア MME と P-GW に向けた制御シグナリングを開始します。シグナリングの場合：

- ベアラーの削除の場合、S-GW は P-GW に Delete-Bearer-Command メッセージを、MME に Delete-Bearer-Request（EBI を使用）メッセージを送信します。
- PDN の削除の場合、S-GW は P-GW に Delete-Session-Request メッセージを、MME に Delete-Bearer-Request（LBI を使用）メッセージを送信します。
- S-GW は、ピアからの削除応答を待機しません。要求は一度のみ送信され、ローカルリソースがリセットされます。

**使用上のガイドライン** このコマンドを使用して、サポートされているいずれかのインターフェイスでパス障害が発生したときに実行するアクションのタイプを指定します。

**例**

次のコマンドは、S5 インターフェイスのパス障害アクションを「signal peer」に設定します。

```
path-failure s5 signal-peer
```

# pco-options

21.1.V0（N5.1）より前のリリースでは、次のようになります。

このコマンドは、ネットワーク内のカスタマイズされたPCO（プロトコル設定オプション）オプションのMS GTP メッセージへの送信を制御し、PCO IE にリンク MTU を含めるようにAPN を設定します。

リリース 21.1.V0（N5.1）以降：

必要に応じて、PCO/APCO/EPCO IE にプロトコル設定オプションを含めるように APN を設定します。

**Command_Product**

P-GW

GGSN

**Command_Privilege**

セキュリティ管理者、管理者

**コマンド モード**

[Exec] > [Global Configuration] > [Context Configuration] > [APN Configuration]

**configure > context** *context_name* **> apn** *apn_name*

上記のコマンドシーケンスを入力すると、次のプロンプトが表示されます。

`[context_name]host_name(config-apn)#`

**構文の説明**

**pco-options { custom1 [ ue-requested ] | link-mtu** *bytes* **[ non-ip** *bytes* **] }**epdg fqdn *domain_name*
**{ default | no } pco-options [ custom1 | link-mtu [ non-ip ]]**

**custom1**

ネットワーク内のカスタマイズされた PCO オプションの MS メッセージへの送信を有効にします。サポートに関係なく、カスタマイズされた PCO オプションをすべての UE に送信します。

**ue-requested**

「UE-Requested」モードの場合、ネットワーク内のカスタマイズされたPCO オプションの MS メッセージへの送信を有効にします。カスタマイズされた PCO オプションを要求する UE のみに PCO を送信します。

**link-mtu *bytes***

21.1.V0（N5.1）より前のリリースでは、次のようになります。

UE に要求された場合、PCO IE にリンク MTU を含めるように APN を設定します。

リリース 21.1.V0（N5.1）以降：

UE に要求された場合、IP および非 IP PDN 接続応答の PCO/APCO/EPCO IE にリンク MTU を含めるように APN を設定します。

最初の接続/スタンドアロンの PDN 接続中に、UE が IPv4 リンク MTU サイズの PCO 要求を送信すると、S-GW/SGSN/HSGW は、セッションの作成要求、PDP コンテキスト要求の作成/更新、またはPBU からP=GW、GGSN、またはPMIP-PGW に、透過的に同じものを送信します。セッション応答の作成、PDP コンテキスト応答の作成または更新、PBA は、APN で設定され

た最新の MTU サイズの PCO 値を使用して送信されます。UE がアウトバウンドローミング中の場合は、デフォルト値（1500）が MTU サイズの PCO で提供されます。

*bytes* は 1280 ～ 2000 の整数である必要があります。

デフォルト：1500

**non-ip *bytes***

非 IP の PDN のリンク MTU です。*bytes* は 128 ～ 2000 の整数である必要があります。デフォルトは 1358 です。

**epdg**

PCO でオペレータ固有の epdg の選択を有効にします。デフォルト設定は無効です。

**fqdn**

完全修飾ドメイン名を指定します。これに基づいて、IP アドレスは DNS からクエリされます。

**default**

ネットワーク内のカスタマイズされた PCO オプションの MS メッセージへの送信を無効にしたり、リンク MTU の PCO を 1500 バイトに設定したりします。

**no**

カスタマイズされた PCO オプションを任意の UE に送信したり、リンク MTU の PCO を 1500 バイトに設定したりしないでください。

## 使用上のガイドライン

このコマンドを使用し、ネットワーク内のカスタマイズされた PCO オプションの MS GTP メッセージの送信を有効または無効にして、リンク MTU サイズの PCO 値を設定します。

**重要**

*ACS* チャージング アクションのコンフィギュレーション モードで **pco-custom1** コマンドのカスタム PCO 値を設定します。

### 例

次のコマンドは、サポートに関係なく、カスタマイズされた PCO オプションのすべての UE への送信を有効にします。

```
pco-options custom1
```

次のコマンドは、ネットワーク内のカスタマイズされた PCO オプションの MS メッセージへの送信を無効にし、リンク MTU PCO を 1500 バイトに設定します。

```
default pco-options
```

次のコマンドは、epdg.com を設定します。

```
pco-options epdg fqdn epdg.com
```

# pdn-type

このコマンドは、APN プロファイルの PDN タイプインジケータを設定するために使用されます。

**Command_Product** MME

**Command_Privilege** セキュリティ管理者、管理者

**コマンド モード** [Exec] > [Global Configuration] > [APN Profile Configuration]

**configure > apn-profile** *profile_name*

上記のコマンドシーケンスを入力すると、次のプロンプトが表示されます。

`[local]host_name(apn-profile-profile_name)#`

**構文の説明**

**pdn-type { ip | non-ip { sgi | t6a [ scef-id** *scef_id* **[ scef-realm** *realm_name* **] ] } }**
**remove pdn-type**

**remove**

キーワードの remove を指定すると、既存の設定が削除されます。

**ip**

セルラー IoT の PDN タイプを IP PDN として設定するには、このキーワードを使用します。

**non-ip**

セルラー IoT の PDN タイプを非 IP PDN として設定するには、このキーワードを使用します。

**sgi**

セルラー IoT 非 IP PDN 配信パスタイプを SGi として設定するには、このキーワードを使用します。

**t6a**

セルラー IoT 非 IP PDN 配信パスタイプを T6a として設定するには、このキーワードを使用します。

**scef-id*scef_id***

ユーザはオプションで、このキーワードを使用して SCEF ID を指定できます。SCEF 識別子は、長さが 1 ～ 63 文字の文字列です。

**scef-realm *realm_name***

オプションでSCEF Diameter レルムを指定するには、このキーワードを使用します。*realm_name* は長さ 1 の文字列で、最大 127 文字です。

**使用上のガイドライン**

セルラー IoT の PDN タイプを指定するには、このコマンドを使用します。このコマンドを使用すると、ユーザは HSS によって提供される APN サブスクリプションの PDN タイプを上書きするオプションを使用できます。このコマンドは、接続および追加の PDN 接続中にのみ適用され、引き渡しのシナリオでは適用されません。このコマンドはデフォルトでは無効になっています。

次のコマンドを使用して、PDN タイプを非 IP および配信パスタイプを SGi として設定します。

**pdn-type non-ip sgi**

次のコマンドを使用して、SCEF 識別子およびレルム名とともに、PDN タイプを非 IP および配信パスを T6a として指定します。

**pdn-type non-ip t6a scef-id** *sc1* **scef-realm** *xyz.com*

# pdp-type

この APN でサポートされている PDP コンテキストのタイプを設定します。

**Command_Product**

GGSN

P-GW

SAEGW

**Command_Privilege**

セキュリティ管理者、管理者

**コマンド モード**

[Exec] > [Global Configuration] > [Context Configuration] > [APN Configuration]

**configure > context** *context_name* **> apn** *apn_name*

上記のコマンドシーケンスを入力すると、次のプロンプトが表示されます。

`[context_name]host_name(config-apn)#`

**構文の説明**

**pdp-type** { **ipv4** [ **ipv6** ] | **ipv6** [ **ipv4** ] | **ppp** | **non-ip** }
**default pdp-type**

**default**

APN のデフォルトの PDP タイプ（IPv4）を設定します。

**ipv4 [ ipv6 ]**

IPv4 PDP コンテキストのサポートを有効にします。IPv6 オプションのキーワードがこのコマンドで入力された場合、IPv6 のサポートも有効にします。デフォルト：有効

重要 いずれかの順序で IPv4 と IPv6 の両方を入力すると、両方のサポートが有効になります。

**ipv6 [ ipv4 ]**

IPv6 PDP コンテキストのサポートを有効にします。また、このコマンドで IPv6 オプションのキーワードが入力されている場合は、IPv4 のサポートも有効にします。デフォルト：[Disabled]

重要 いずれかの順序で IPv4 と IPv6 の両方を入力すると、両方のサポートが有効になります。

**ppp**

PPP PDP コンテキストのサポートを有効にします。デフォルト：[無効（Disabled）]

**non-ip**

APN の非 IP PDP タイプのサポートを有効にします。

**使用上のガイドライン** IP PDP コンテキストタイプとは、MS が IP を使用してインターネットまたはイントラネットなどの PDN と通信するものです。PPP PDP コンテキストは、MS からの PPP または PPP Network Control Protocol（NCP; ネットワーク コントロール プロトコル）のフレームが、GGSN で終端されるか、または GGSN によって転送されるかのいずれかです。

セッションで APN でサポートされていない PDP タイプが指定されている場合、システムは原因コード 220（DCH、不明な PDP アドレス、または PDP タイプ）でセッションを拒否します。

注意 IPv6 コールが機能するには、接続先コンテキストに少なくとも 1 つの IPv6 インターフェイスが設定されている必要があります。

**例**

次のコマンドは、PPP コンテキストタイプをサポートするように APN を設定します。

```
pdp-type ppp
```

# psm

このコマンドは、UE 省電力モードのパラメータを設定するために使用されます。

**Command_Product** MME

**Command_Privilege** セキュリティ管理者、管理者

**コマンド モード** [Exec] > [Global Configuration] > [Call Control Profile Configuration]

**configure > call-control-profile** *profile_name*

上記のコマンドシーケンスを入力すると、次のプロンプトが表示されます。

`[local]host_name(config-call-control-profile-profile_name)#`

**構文の説明**

**[remove] psm {ue-requested [dl-buf-duration [packet-count** *packet_value* **]]| t3324-timeout** *t3324_value* **t3412-extended-timeout** *t3412_ext_value* **[dl-buf-duration [packet-count** *packet_value* **]]}**

**remove**

**remove** キーワードを指定すると、既存の省電力モードの設定が削除されます。

**ue-requested**

このキーワードは、アクティブタイマーと拡張定期タイマーの UE 要求値を受け入れる場合に使用します

**t3324-timeout *t3324_value***

T3324 のアクティブタイマー値を設定するには、このキーワードを使用します。

*t3324_value*

T3324 のアクティブタイマーは、0 ～ 11160 秒の範囲の整数値です。

**t3412-extended-timeout *t3412_ext_value***

T3412 の拡張タイマー値を設定するには、このキーワードを使用します。

*t3412_ext_value*

T3412 の拡張タイマーは、0 ～ 35712000 秒の範囲の整数値です。

**dl-buf-duration**

このキーワードは、UE をページングできない場合に DDN ACK 内でダウンリンクバッファ期間を送信するために使用します。

**packet-count *packet_value***

このキーワードは、UE をページングできない場合に、DDN ACK 内で「DL Buffering Suggested Packet Count」を送信するために使用します。

*packet_value*

*packet_value* は、0 ～ 65535 の整数値です。

## 使用上のガイドライン

このCLIコマンドを使用して、T3324のアクティブタイマーおよびT3412の拡張タイマーを設定します。CLIでは、これらのタイマーに対して、UE要求値またはHSS登録値またはMME設定値を受け入れるオプションも提供しています。このコマンドは、DDN Ackでのダウンリンクバッファ期間、 DDN Ackオプションの IE「ダウンリンクの推奨パケット数」を送信するかどうかを設定するために使用します。CLIオプションの **dl-buf-duration [ packet-count** *packet_value* **]** は、DDN Ackでのダウンリンクバッファ期間を送信するかどうかをオプションで設定するために使用されます。また、DDN AckオプションのIE「ダウンリンクの推奨パケット数」も設定できます。このオプションが設定されておらず、サブスクリプションで送信されていない場合、MMEはDDN拒否でIEを送信しません。**packet-count** の値がローカルに設定されていない場合は、**packet-count** のサブスクリプション値が使用されます。サブスクリプション値は「0」にすることができます。この場合、ローカルで設定されていても、そのサブスクライバに対してパケットカウントIEは送信されません。T3324のアクティブタイマーおよびT3412の拡張タイマーがローカルに設定されている場合、これらの値は常に使用されます。**psm** コマンドがアクティブタイマーと拡張定期タイマーに対してUE要求値を使用するように設定されている場合は、UE要求値は受け入れられますが、UEがT3412の拡張タイマーを要求しない場合は、サブスクリプションデータで使用可能な値が拡張定期タイマーに使用されます。サブスクリプションデータで値が使用できない場合は、MMEサービスで設定された値が使用されます。

3GPP TS 24.008の最新バージョンでは、T3412の拡張タイマーの最大値は「320 X 31」時間、つまり「35712000」秒になります。タイマーの実装におけるMMEの制約により、T3412の拡張タイマーは1050時間、つまり「3780000」秒に制限されています。ただし、3GPP TS 24.008 GPRS Timer 3としてこのタイマーに最も近い使用可能な値は、960時間（320 X 3）、つまり3456000秒です。

### 例

次のコマンドを使用して、省電力モードを有効にし、T3324タイマーおよびT3412タイマーのUE要求値を受け入れるようにします。

```
psm ue-requested
```

次のコマンドを使用して、UE省電力モードを有効にし、T3324タイマーおよびT3412タイマーにオペレータが望ましい値を指定します。

```
psm t3324-timeout 100 t3412-extended-timout 5000
```

次のコマンドを使用して、 PSMを有効にし、T3324タイマーおよびT3412タイマーのUE要求値を受け入れるようにします。また、このコマンドは、UEをページングできない場合に、DDN ACK内の「DLバッファリングの推奨パケット数」も指定します。

```
psm ue-requested dl-buf-duration packet-count 100
```

次の例では、PSMが有効になっており、T3324タイマーおよびT3412タイマーの値がDDN ACK内のパケットカウントの設定とともに指定されています。

```
psm t3324-timeout 1000 t3412-extended-timeout 5000 dl-buf-duration
packet-count 100
```

# require session ipsecmgr-per-vcpu

vCPU ごとの IP セキュリティマネージャ（ipsecmgr）プロセスの数を設定します。

**Command_Product** ePDG（VPC-DI プラットフォームのみ）

**Command_Privilege** セキュリティ管理者、管理者

**コマンド モード** [Exec] > [Global Configuration]

**configure**

上記のコマンドシーケンスを入力すると、次のプロンプトが表示されます。

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

**構文の説明** **[ default ] require session ipsecmgr-per-vcpu** *count* **}**

**default**

vCPU ごとの ipsecmgrs の数をデフォルトの 1 に戻します。

（注） デフォルト値は、コールモデルの導入要件に応じて必要に応じて調整できます。詳細については、販売またはサポート担当者にお問い合わせください。

***count***

各 vCPU に対して作成される ipsecmgr プロセスの 1 ～ 2 の番号を設定します。デフォルト：1。

**使用上のガイドライン** vCPU ごとに複数の IP セキュリティマネージャ（ipsecmgr）プロセスを有効にします。

**例**

次のコマンドは、vCPU ごとに 2 つの ipsecmgrs を作成するようにシステムを設定します。

```
require session ipsecmgr-per-vcpu 2
```

# require session sessmgr-per-vcpu

vCPU ごとにセッションマネージャ（sessmgr）プロセスの数を設定します。

**Command_Product** すべて（VPC-DI プラットフォームのみ）

**Command_Privilege** セキュリティ管理者、管理者

**コマンド モード** [Exec] > [Global Configuration]

**configure**

上記のコマンドシーケンスを入力すると、次のプロンプトが表示されます。

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

**構文の説明** **[ default ] require session sessmgr-per-vcpu** *count* **}**

**default**

vCPU ごとのセッションマネージャの数をデフォルトの 1 に戻します。

⚠

**注意** デフォルト値は、コールモデルの導入要件に応じて必要に応じて調整できます。ただし、デフォルト値を使用することを推奨します。デフォルト値を変更または調整するには、営業担当者またはサポート担当者にお問い合わせください。

***count***

vCPU ごとに作成されるセッション マネージャ プロセスの数を設定します。有効な値は 1 と 2 です。デフォルト値は 1 です。

MME/SGSN の場合のみ、vCPU あたりのセッションマネージャの数は 2 になります。

他のすべての値は予約されています。

**使用上のガイドライン** CPU の使用率は低いものの、RAM の使用率が高いアプリケーション（Internet of Things（IoT）など）の場合、vCPU ごとに複数のセッションマネージャ プロセスを設定するほうが効率的です。

**表 *39 :* プラットフォームごとの *vCPU* サポート**

| プラットフォーム | **vCPU サポート** |
|---|---|
| Gateway | • vCPU あたり 1 つのセッション マネージャ プロセスの場合、サービス機能（SF）VM あたり 16 のセッションマネージャ プロセスがサポートされます。<br>• vCPU あたり 2 つの セッション マネージャ プロセスの場合、SF VM あたり 32 の セッション マネージャ プロセスがサポートされます。 |
| MME/SGSN | vCPU あたり 2 つの セッション マネージャ プロセスの場合、SF VM あたり 56 の セッション マネージャ プロセスがサポートされます。 |

### 例

次のコマンドは、vCPU ごとに 2 つのセッション マネージャを作成するようにシステムを設定します。

```
require session sessmgr-per-vcpu 2
```

# scef-service

このコマンドは、SCEF サービスをコール制御プロファイルに関連付けます。

**Command_Product**

MME

**Command_Privilege**

管理者

**コマンド モード**

[Exec] > [Global Configuration] > [Call Control Profile Configuration]

**configure > call-control-profile** *profile_name*

上記のコマンドシーケンスを入力すると、次のプロンプトが表示されます。

`[local]host_name(config-call-control-profile-profile_name)#`

### 構文

**[ remove ] associate scef-service** *service_name*

#### remove

このコマンドプレフィックスは、コール制御プロファイルから SCEF の関連付けを削除します。

#### associate

このコマンドは、コール制御プロファイルに SCEF サービスを関連付けます。

#### scef-service*service_name*

このコマンドは、サービス名で識別されるコール制御プロファイルに SCEF を関連付けます。サービス名は 1 ～ 63 の文字列です。

**使用上のガイドライン**

このコマンドを使用して、非 IP データ配信（NIDD）のコール制御プロファイルに SCEF サービスを関連付けます。

# scef-service

このコマンドは、SCEF-service を MME サービスに関連付けます。

**Command_Product** MME

**Command_Privilege** 管理者

**コマンド モード** [Exec] > [Global Configuration] > [Context Configuration] > [MME Service]

**configure > context** *context_name* **> mme-service** *service_name*

上記のコマンドシーケンスを入力すると、次のプロンプトが表示されます。

`[context_name]host_name(config-mme-service)#`

**構文**

**[ remove ] associate scef-service** *service_name*

**remove**

このコマンドプレフィックスは、MME サービスから SCEF アソシエーションを削除します。

**associate**

このコマンドは、SCEF サービスを MME サービスに関連付けます。

**scef-service*service_name***

このコマンドは、サービス名で識別される MME サービスに SCEF を関連付けます。サービス名は 1 ～ 63 の文字列です。

**使用上のガイドライン** このコマンドを使用して、非 IP データ配信（NIDD）の MME サービスに SCEF サービスを関連付けます。

# serving-plmn-rate-control

このコマンドは、コントロールプレーン CIoT 最適化のためのサービス PLMN レートコントロールを設定するために使用されます。サービス提供レート制御によって、CP 最適化が有効になっている場合に、そのコントロールプレーンを介して、または PGW/SCEF がデータを送信できるレートが制限されます。

**Command_Product** MME

**Command_Privilege** セキュリティ管理者、管理者

**コマンド モード** [Exec] > [Global Configuration] > [Call Control Profile Configuration]

**configure > call-control-profile** *profile_name*

上記のコマンドシーケンスを入力すると、次のプロンプトが表示されます。

`[local]host_name(config-call-control-profile-profile_name)#`

構文の説明

**serving-plmn-rate-control ul-rate** *ul_rate_value* **dl-rate** *dl_rate_value*
**remove serving-plmn-rate-control**

**remove**

キーワードの remove を指定すると、既存の設定が削除されます。

**ul-rate*ul_rate_value***

deci 時間（6 分）ごとに UE がアップリンクパスでが送信できるデータ NAS PDU の最大数。アップリンクレートは 10 ～ 65535 の整数です。この場合、65535 の値は、deci 時間ごとに UE がアップリンクパスで送信できる PDU の数に制限がないことを意味します。

**dl-rate*dl_rate_value***

PGW/SCEF が deci 時間（6 分）ごとにダウンリンクパスで UE に送信できるデータ NAS PDU の最大数。ダウンリンクレートは 10 ～ 65535 の整数です。この場合、65535 の値は、PGW/SCEF が deci 時間あたりにダウンリンクパスで送信できる PDU の数に制限がないことを意味します。

使用上のガイドライン

このコマンドは、NAS を介したデータのサービス PLMN レートを設定します。コントロールプレーン CIoT の最適化を使用している間、UE と PGW/SCEF 間のデータ交換のレートを制限します。このコマンドはデフォルトでは無効になっています。

**例**

次のコマンドを使用して、アップリンクレートを 35、ダウンリンクレートを 45 に設定し、NAS 経由のデータに対するサービス PLMN レートを設定します。

**serving-plmn-rate-control ul-rate** *35* **dl-rate** *45*

# show card

**show card table** コマンドと **show card information** コマンドの出力は、新しいスロットタイプを反映するように変更されました。

サービス機能（SF）、ネットワーク機能（NF）、およびアプリケーション機能（AF）は、総称して機能カードとして表現され、これらのコマンドの出力では共通のスロットタイプ「FC」の下にグループ化されます。

以前は、これらのカードは次のスロットタイプで表されていました。

- サービス機能：SFC
- ネットワーク機能：NFC
- アプリケーション機能：AFC

（注） スロット 1 と 2 に常駐する制御機能（CF）仮想カードタイプのスロットタイプ指定子は、「CFC」のままです。

次に、新しい **show card table** コマンドと **show card information** コマンドの出力例を示します。

### show card table の例

```
show card table
```

```
1: CFC      Control Function Virtual Card      Standby        -
2: CFC      Control Function Virtual Card      Active         No

3: FC  2-Port Service Function Virtual Card    Standby        -
8: FC  1-Port Network Function Virtual Card    Active         No
9: FC  1-Port Application Func. Virtual Card   Active         No
10:FC  1-Port Service Function Virtual Card    Standby        -
```

### show card information の例

```
show card information 3
```

```
Card 3:
  Slot Type               : FC
  Card Type               : 2-Port Service Function Virtual Card
  Operational State       : Standby
  Desired Mode            : Standby
```

# show cloud configuration

設定ファイルの内容を表示します。

### 製品

すべて

### 権限

セキュリティ管理者、管理者、インスペクタ、オペレータ

### モード

Exec

次のプロンプトが Exec モードで表示されます。

```
[local]host_name#
```

### 構文

**show cloud configuration**

### 使用法

このコマンドは、設定ファイルの内容を画面にダンプします。設定ファイルを設定ディスクまたはローカルフラッシュ上に表示します。通常、ユーザはこれらのファイルに直接アクセスすることはできません。フラッシュ上のローカル パラメータ ファイルは VPC のインストール時に定義され、設定ディスクは通常、オーケストレータによって作成されてからカードに接続されます。

### 例

このコマンドは、カード番号1に関連付けられているハードウェア設定を表示します。

**show cloud configuration**

# show cloud hardware

各カードまたは特定のカードのハードウェア設定を表示します。

### 製品

すべて

### 権限

セキュリティ管理者、管理者、インスペクタ、オペレータ

### モード

Exec

次のプロンプトが Exec モードで表示されます。

```
[local]host_name#
```

### 構文

**show cloud hardware** *card_number*

#### *card_number*

情報を表示するカードの番号を指定します。

### 使用法

VPC 内の特定のカードまたはすべてのカードの基盤となる VM ハードウェアの設定を表示します。設定されている vCPU、メモリサイズ、HugePage サイズ、暗号化ハードウェア、および NIC に関する情報を表示します。

### 例

このコマンドは、カード番号 1 に関連付けられているハードウェア設定を表示します。

**show cloud hardware 1**

# show cloud hardware optimum

一覧表示されたハードウェアパラメータで最高のスループットを得るために最適なハードウェア設定を表示します。

### 製品

すべて

### 権限

セキュリティ管理者、管理者、インスペクタ、オペレータ

### モード

Exec

次のプロンプトが Exec モードで表示されます。

```
[local]host_name#
```

### 構文

**show cloud hardware optimum**

### 使用法

使用可能なパラメータに応じて、基盤となる VM ハードウェアの最適な設定を表示します。設定されている vCPU、メモリサイズ、HugePage サイズ、暗号化ハードウェア、および NIC に関する情報を表示します。

### 例

このコマンドは、関連付けられた VM ハードウェアに最適なハードウェア設定を表示します。

**show cloud hardware optimum**

# show cloud hardware test

各カード、または特定のカードの現在のハードウェア設定を最適な設定と比較します。

### 製品

すべて

### 権限

セキュリティ管理者、管理者、インスペクタ、オペレータ

### モード

Exec

次のプロンプトが Exec モードで表示されます。

```
[local]host_name#
```

### 構文

**show cloud hardware test** *card_number*

#### *card_number*

情報を表示するカードの番号を指定します。

### 使用法

特定のカードまたは VPC 内のすべてのカードの基盤となる VM ハードウェアの設定を最適な設定と比較します。設定されている vCPU、メモリサイズ、HugePage サイズ、暗号化ハードウェア、および NIC に関する情報を表示し、パラメータごとの最適値を示します。

### 例

このコマンドは、カード番号1に関連付けられているハードウェア設定を表示します。

```
show cloud hardware test 1
```

# show cloud monitor

VPC 内のすべてのカードまたは特定のカードの VPC-DI のネットワーク遅延とパケット損失の統計情報を表示します。

### 製品

すべて

### 権限

セキュリティ管理者、管理者、インスペクタ、オペレータ

**コマンド モード**

Exec

次のプロンプトが Exec モードで表示されます。

```
[local]host_name#
```

**構文の説明**

**show cloud monitor di-network** {**detail** | **summary**} *card_number*

**detail**

VPC-DI ネットワークに関する詳細情報を表示します。

**summary**

VPC-DI ネットワークに関する要約情報を表示します。

***card_number***

情報を表示するカードの番号を指定します。

**使用上のガイドライン**

VPC 内の特定のカードまたはすべてのカードの基盤となる VM ハードウェアの設定を表示します。設定されている vCPU、メモリサイズ、HugePage サイズ、暗号化ハードウェア、および NIC に関する情報を表示します。

### 例

このコマンドは、VPC 内の 3 番目のカードとの間の VPC-DI ネットワーク通信でのモニタ対象統計情報の要約を表示します。この表示には、過去 5 分間および過去 60 分間のテストパケット損失率が示されます。この率が 1% を超えると、正常性ステータスが「Bad」としてマークされます。

**show cloud monitor di-network summary 3**

```
Card 3 Test Results:

ToCard   Health   5MinLoss   60MinLoss
   1      Good       0.0%       0.0%
   2      Good       0.0%       0.0%
   4       Bad       6.32%      5.36%
   5      Good       0.0%       0.0%
   6      Good       0.0%       0.0%
```

# show scef-service statistics

SCEF サービスの設定およびステータス情報を表示します。

**Command_Product**

MME

**Command_Privilege** セキュリティ管理者、管理者、オペレータ、インスペクタ

**コマンド モード** Exec

次のプロンプトが Exec モードで表示されます。

`[local]host_name#`

**構文の説明** **show scef-service statistics { all | name** *service_name* **| summary }**

**all**

すべての SCEF サービスのすべての使用可能な設定およびステータス情報を表示します。

**name *service_name***

指定された SCEF サービス名のステータス情報を表示します。

**summary**

使用可能な SCEF サービスの統計情報の要約を表示します。

**使用上のガイドライン** SCEF サービス情報とその統計情報を表示するには、このコマンドを使用します。

**例**

次のコマンドは、すべての SCEF サービスの統計情報を表示します。

**show scef-service statistics all**

次のコマンドは、サービス名 *Test* を使用した SCEF サービス設定の情報を表示します。

**show scef-service statistics name** *Test*

**重要** コマンドの出力の説明は、『*Statistics and Counters Reference*』に記載されています。

# show system ssh key status

現在使用中の内部 SSH キーのフィンガープリント、キーが検出された送信元、およびすべてのオンライン VM のステータスを表示します。

**Command_Product** VPC-DI

**Command_Privilege** セキュリティ管理者、管理者、オペレータ

**コマンド モード** Exec

次のプロンプトが Exec モードで表示されます。

`[local]host_name#`

構文の説明

**show system ssh key status [ | { grep** *grep_options* **| more } ]**

**|{ grep *grep_options* | more }**

コマンドの出力をパイプ（送信）して、指定したコマンドを実行します。出力の送信先となるコマンドを指定する必要があります。

**grep**コマンドおよび**more**コマンドの使用方法について詳しくは、「コマンドライン インターフェイスの概要」の章の「コマンドの出力の制御」の項を参照してください。

使用上のガイドライン

このコマンドは、リモートコマンドの実行やファイル転送など、VPC-DI システム内のすべてのコンポーネント VM 間の内部通信に使用される SSH キーに関する情報を表示します。

# system packet-dump

VPC-DI システムの SF カードまたは CF カード上でのパケットダンプを開始します。

**Command_Product**

すべて

**Command_Privilege**

セキュリティ管理者、管理者

コマンド モード

Exec

次のプロンプトが Exec モードで表示されます。

```
[local]host_name#
```

構文の説明

**system packet-dump { di-net card** *slot_num* **| port** *service_port* **} [ bond { a | b } | direction { both-rxtx | rx | rxtx | tx } | duration** *seconds* **| packet-type { ipv4 | ipv6 } | pcapfile-size** *size* **| pcapfile-split-val** *value* **| protocol { icmpv4 | icmpv6 | tcp | udp } | to file** *filename* **]**

**di-net card *slot_num***

1 ～ *n* のカードを指定します。

**port *card_port/port_num***

1 ～ *n* のカード番号と 1 ～ 50 のポート番号に基づいてイーサネット インターフェイスを指定します（3/1 など）。

**bond { a | b }**

結合インターフェイスのスレーブを指定します。

**direction { both-rxtx | rx | rxtx | tx }**

キャプチャするパケットの方向のフィルタを、受信（**rx**）、送信（**tx**）、または両方（**rxtx**）のいずれかに指定します。受信と送信の両方をキャプチャするには **both-rxtx** オプションを使用しますが、それぞれが別のファイルに出力されます。

**duration *seconds***

パケットダンプの秒数を 1 ～ 600 で指定します。デフォルト：5 秒

**packet-type { ipv4 | ipv6 }**

キャプチャするパケットのタイプのフィルタを **ipv4** または **ipv6** に指定します。

**pcapfile-size *size***

各パケットキャプチャ（pcap）ファイルの最大サイズを 10 ～ 800 メガバイトで指定します。デフォルトは 10 メガバイトです。

**pcapfile-split-val *value***

指定されたキャプチャに対して生成する pcap ファイルの数を 0 ～ 10 で指定します。デフォルト：0（ファイルを分割しません）。

**protocol { icmpv4 | icmpv6 | tcp | udp }**

キャプチャするパケットのプロトコルを **icmpv4**、**icmpv6**、**tcp**、または **udp** に指定します。

**to file { /flash | /hd-raid | /cdrom1 | /sftp } [/directory]/ *filename***

出力の場所とファイル名を指定します。

**使用上のガイドライン**

このコマンドを使用してパケットキャプチャを実行し、VPC-DI展開内の問題をトラブルシューティングします。

**例**

次のコマンドは、スロット 7、ポート 1 のカードでパケットダンプを開始し、/flash/example7-1.pcap でローカルに保存されたファイルにダンプを出力します。

```
system packet-dump port 7/1 to file /flash/example7-1.pcap
```

# system ping

VPC-DI システム内の 2 つの VM 間の内部ネットワーク上で ping テストを開始します。

**Command_Product**

VPC-DI

**Command_Privilege**

セキュリティ管理者、管理者

**コマンド モード**

Exec

次のプロンプトが Exec モードで表示されます。

```
[local]host_name#
```

構文の説明

**system ping from card** *slot_num* **to card** *slot_num* **[ count** *number_of_packets* **| size** *bytes* **]**

**from card *slot_num***

ping テストの発信元となる、カードスロット番号を 1 ～ *n* で指定します。

**to card *slot_num***

接続先のカードスロット番号を 1 ～ *n* で指定します。

**count *number_of_packets***

送信される ping パケットの数を 1 ~ 10000 で設定します。デフォルト：5 パケット

**size *bytes***

ICMP データグラムのサイズを 40 ～ 18432 のバイト単位で設定します。デフォルト：56

使用上のガイドライン

このコマンドを使用して ping テストを実行し、VPC-DI 展開内の接続の問題をトラブルシューティングします。

例

次に、スロット 1 のカードからスロット 9 のカードへの 1,000 個のパケットの ping テストを開始するコマンドを示します。

```
system ping from card 1 to card 9 count 1000
```

# system ssh

VPC DI システム内のカード（VM）間の内部 SSH セッションで使用される永続的な SSH ユーザキーを管理します。

**Command_Product**

VPC-DI

**Command_Privilege**

Security Administrator

コマンド モード

Exec

次のプロンプトが Exec モードで表示されます。

```
[local]host_name#
```

構文の説明

**system ssh key { copy boot1 to card** *slot_num* **| create boot1 }**

**no system ssh key boot1 { all | card** *slot_num* **}**

**no system ssh key boot1 { all | card *slot_num* }**

VPC-DI システム内の特定のカードまたはすべてのカードの永続的な SSH キーを削除します。キーの削除は、永続的なキーの VM を消去したり、別の配信方式（ESC、OpenStack、添付された ISO）を使用するようにシステムを準備したりするために使用できます。

- **all** ：VPC-DI システム内のすべてのカードの SSH キーを削除します。
- **card** *slot_num*：*slot_num* によって指定されたカードの SSH キーを削除します。

（注） このコマンドは、再起動されるまでVMに影響を与えません。ブート中に検出されたアクティブキーを引き続き使用します。

**copy boot1 to card *slot_num***

アクティブな CF の /boot1 で、永続的な SSH キー（パブリックとプライベートの両方）を別の VM に転送します。この VM は、起動に失敗したときに、コンソールアクセス権を持つユーザが、その VM を受信側モードにして受け入れるようにする必要があります。

**create boot1**

新しい永続的 SSH キー（パブリックとプライベートの両方）を作成し、アクティブ CF 上の /boot1 に保存します。

（注） このコマンドは、再起動されるまでVMに影響を与えません。ブート中に検出されたアクティブキーを引き続き使用します。

## 使用上のガイドライン

このコマンドを使用して、 VPC-DI の展開で内部 SSH キーペアを管理します。StarOS は、管理 VM（CF）上のユーザ CLI および SFTP セッションに SSHD サービスを提供しますが、リモートコマンドの実行やファイル転送など、すべてのコンポーネントVM間の内部通信専用にするために、別の SSHD サービスセットを実行します。この内部 SSHD は、内部の DI ネットワークインターフェイスでのみ使用されます。

このコマンドを使用すると、VM の仮想ディスクドライブ（HDD）の SSH キーを保存および管理できます。これにより、Cisco Elastic Services Controller（ESC）、OpenStack、または ISO の直接接続といった方法に加えて、SSHキーペアを保存するための代替オプションが提供されます。/boot1 パーティションには、セキュリティ管理者のみがアクセスできます。

**show system ssh key status**現在使用中の公開キーのフィンガープリント、キーが検出された送信元、およびすべてのオンラインVMのステータスを表示するには、コマンドを使用します。

### 例

次のコマンドでは、SSH キーペアをアクティブ CF からスロット 12 のカードにコピーします。

```
system ssh key copy boot1 to card 12
```

# tunnel udpip

APN の P-GW と外部アプリケーションサーバ間の UDP-IPv4 または UDP-IPv6 のトンネリングパラメータを設定します。

**Command_Product**

P-GW

S-GW

**Command_Privilege**

セキュリティ管理者、管理者

**コマンド モード**

[Exec] > [Global Configuration] > [Context Configuration] > [APN Configuration]

**configure > context** *context_name* **> apn** *apn_name*

上記のコマンドシーケンスを入力すると、次のプロンプトが表示されます。

`[context_name]host_name(config-apn)#`

**構文の説明**

**tunnel udpip peer-address** *peer_address* **peer-port** *peer_udp_port* **[ local-port** *local_udp_port* **]**
**no tunnel udpip**

**no**

APN の UDP IPv4 または UDP-IPv6 のトンネリングを無効にします。

**peer-address *peer_address***

トンネルのピアアドレスを指定します。

*peer_address* はドット付き 10 進表記で表す必要があります。

**peer-port *peer_udp_port***

トンネルのピアのポート番号を指定します。

*peer_udp_port* はドット付き 10 進表記で表す必要があります。

**local-port *local_udp_port***

ローカル UDP ポート番号を指定します。

デフォルト：49152

**使用上のガイドライン**

ローカルおよびピアの UDP ポート番号については、IANA に未登録のポート番号を使用することを推奨します。

この CLI コマンドは、APN への S5/S8 インターフェイス上での新しいサブスクライバコールの作成時に有効になります。

### 例

次に、UDP-IPv4 を使用してサブスクライバトラフィックをカプセル化し、ポート番号 *49152* のローカルに割り当てられた IP アドレスからピア UDP ポート *11220* 上の IP アドレス *192.168.1.100* の外部アプリケーションサーバにトンネリングするようにシステムを設定するコマンドを示します。

```
tunnel udpip peer-address 192.168.1.100 peer-port 11220 local-port 49152
```

VPC コマンド