



NETCONF および ConfD

この章では、NETCONF および ConfD manager と呼ばれる StarOS プロセスについて説明します。

ここで説明する内容は、次のとおりです。

- [機能の概要と変更履歴 \(1 ページ\)](#)
- [概要 \(3 ページ\)](#)
- [ConfD の設定 \(4 ページ\)](#)
- [設定の確認 \(10 ページ\)](#)
- [YANG モデル \(17 ページ\)](#)
- [Show Support Details \(SSD\) \(18 ページ\)](#)
- [ConfD の例 \(18 ページ\)](#)
- [ConfD のアップグレードサポート \(24 ページ\)](#)
- [ECS コマンド用の CLI ベースの YANG モデル \(24 ページ\)](#)

機能の概要と変更履歴

要約データ

該当製品または機能エリア	すべて
該当プラットフォーム	ASR 5500 VPC-SI VPC-DI
機能のデフォルト	無効：設定が必要
このリリースでの関連する変更点	N/A

関連資料	<ul style="list-style-type: none"> • <i>ASR 5500 System Administration Guide</i> • <i>Command Line Interface Reference</i> • VPC-DI システム管理ガイド • 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
<p>ノード選択とロードバランシング (NSLB) の主要業績評価指標 (KPI) のキャプチャに対するサポートが追加されました。</p> <p>この機能を有効にし、これらの KPI を収集するために使用する間隔を設定するために NETCONF プロトコル コンフィギュレーション モードの kpi コマンドが追加されました。</p> <p>詳細については、kpi (6 ページ) および show confdmgr コマンド (11 ページ) の項を参照してください。</p>	21.6
<p>SNMP MIB アラートとアラームは、NETCONF 通知を介して送信できるようになりました。</p> <p>この機能を有効にするために、NETCONF プロトコル コンフィギュレーションモードの netconf コマンドに snmp キーワードが追加されました。</p> <p>show confdmgr コマンド出力が拡張されました。</p>	21.3
<p>ConfD で、REST インターフェイスを介して取得されたバルク統計の運用データを収集できるようになりました。</p> <p>新しい StarOS バルク統計情報とサーバー ConfD 設定の YANG モデルがサポートされています。StarOS CLI を介したすべての更新が自動的に ConfD データベースに同期されます。CLI ベースの YANG モデルは、StarOS ECS (Enhanced Charging System) コマンドにのみ適用されます。</p> <p>NETCONF プロトコル コンフィギュレーションモードに bulkstats、netconf、および rest コマンドが追加されました。autosave-config コマンドは廃止されました。</p> <p>show confdmgr コマンドにキーワード model bulkstats と model confd が追加されました。</p> <p>show confdmgr コマンド出力が拡張されました。</p>	21.2

改訂の詳細	リリース
最初の導入。	21.2 よりも前

概要

StarOS は、Cisco Network Service Orchestrator (NSO) とともに設定および運用データを転送するための YANG データモデルをサポートするノースバウンド NETCONF インターフェイスを提供します。また、NSO 管理コンソールと通信するための ConfD マネージャ (confdmgr) も組み込まれています。

NETCONF (Network Configuration Protocol) は、IETF (RFC 6241) によって開発および標準化されたネットワーク管理プロトコルです。ネットワークデバイスの設定をインストール、操作、削除するためのメカニズムを提供します。この操作は、単純なリモートプロシージャコール (RPC) レイヤの最上位で実現されます。NETCONF プロトコルは、設定データとプロトコルメッセージに XML ベースのデータ符号化を使用します。プロトコルメッセージは、セキュアなトランスポートプロトコル上で交換されます。

ConfD は、デバイスを管理するための一連のインターフェイスを提供する、オンデバイス管理フレームワークです。ConfD フレームワークは、データモデルからすべての管理インターフェイスを自動的にレンダリングします。ConfD は完全な NETCONF 仕様を実装し、XML でエンコードされたコンテンツを使用して SSH で実行します。

ConfD は、外部認証を介して認証/認可されたアクセスのみを許可するように設定されています。confdmgr は、新しいユーザーを認証/認可するときに、ConfD が起動するためのスタンドアロン CLI モジュールを提供します。ConfD は、StarOS 認証を介して認可されたアクセスのみを許可するように設定されています。認証時に、ユーザーには権限レベル (0~15) が与えられます。この権限レベルは、YANG モデルで定義されているように、StarOS *secure admin*、*admin*、*operator*、および *inspector* にマッピングされます。StarOS は、各 ConfD 認証要求の CLI 認証イベントやステータスメッセージをログに記録します。

サウスバウンド側では、ConfD は、ConfD 管理エージェントによって提供される一連の API を介して呼び出された StarOS プロセスと通信します。ConfD コンフィギュレーションデータベース (CDB) は、オブジェクトを保存するために ConfD によって使用されます。StarOS は、ConfD 提供の API を使用してデータベースにアクセスします。StarOS CLI を介したすべての更新が自動的に CDB に同期されます。

YANG は、NETCONF ネットワーク コンフィギュレーションプロトコル向けのデータモデリング言語です。これは、設定データとネットワーク要素の状態データの両方をモデル化するために使用できます。YANG は、ネットワーク要素によって生成されるイベント通知の形式を定義するために使用することもできます。また、データモデルは、NETCONF プロトコル (RFC 6020) を介してネットワーク要素で呼び出すことができるリモートプロシージャコールの署名を定義することができます。YANG ファイルは StarOS の一部としてコンパイルされ、既存の StarOS でサポートされている CLI コマンドを組み込みます。

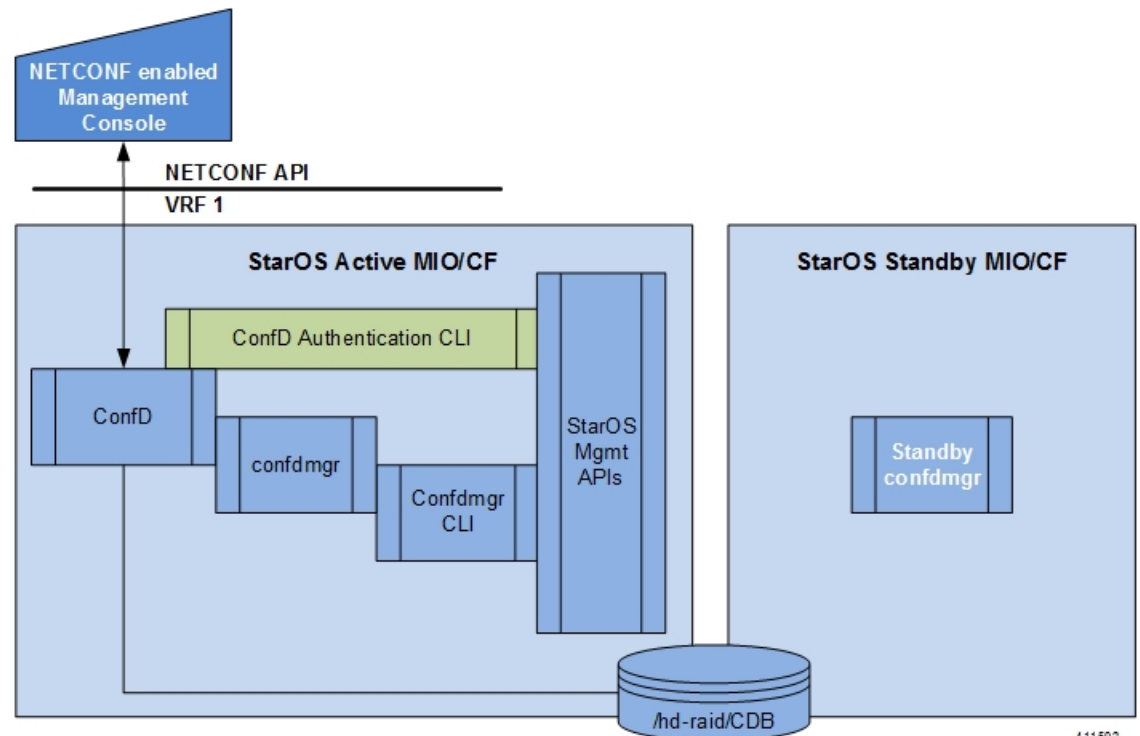
ConfD は、バルク統計情報の運用データを収集することもできます。有効にすると、StarOS は統計情報の収集時に confdmgr にスキーマ情報を送信します。収集されたバルク統計情報は、

REST (Representational State Transfer) インターフェイスを介して後で取得するために、ConfD CDB に保存されます。RESTCONF はトランスポートとして HTTP を使用して、YANG 仕様を RESTful インターフェイスにマッピングする方法を説明する IETF ドラフト

(draft-bierman-netconf-restconf-4) です。REST および RESTCONF は、有効な証明書とキーが設定されている場合にのみ内部で有効になります。クライアント認証が有効になっている場合は、CA 証明書も必要になることがあります。

NSO の詳細については、NSO のユーザーマニュアルを参照してください。

図 1: NETCONF システムフロー



ConfD の設定

StarOS で NETCONF プロトコルを有効にするには、**server confd** を有効にして、Netconf プロトコル コンフィギュレーション モードを開始する必要があります。NETCONF プロトコル コンフィギュレーションモードでは、オプションのコンフィギュレーションコマンドがサポートされています。

SSH キーの要件

NETCONF-ConfD のサポートでは、ローカルコンテキストで V2-RSA SSH キーが設定されている必要があります。

SSH キーが使用できない場合、StarOS はエラーメッセージを生成します。

Failure: The ConfD (NETCONF) server requires an RSA key on the local context

Show ssh key コマンドを実行して、システム上の SSH キーが存在することを確認できます。

SSH キーが使用できない場合は、このガイドの「使用する前に」の章の「SSH オプションの設定」の項を参照してください。

NETCONF プロトコル コンフィギュレーション モード

NETCONF プロトコルは、コンテキスト コンフィギュレーション モードの **erver conf** コマンドを介して有効にします。このコマンドは、ローカルコンテキストのみに制限されます。

```
[local]host_name# configure
[local]host_name(config)# context local
[local]host_name(config-ctx)# server confd
[local]host_name(config-confd)# ?

bulkstats          - Populate ConfD with bulkstats operational data
confd-user         - Configures the default login user with full administrator rights
                    for the ConfD server.
do                 - Spawns an exec mode command which displays
                    information to the administrator
end                - Exits configuration mode and returns to Exec Mode
exit               - Exits current configuration mode, returning to previous mode
kpi                - Key performance indicators gathering interval
netconf            - Configure the netconf interface
no                 - Enables/Disables the followed option
rest               - Configure the rest interface
```

次のキーワードはオプションです。

- **bulkstats**
- **confd-user**
- **kpi**
- **netconf**
- **rest**

NETCONF プロトコルを無効にするには、コンテキスト コンフィギュレーション モードの **no server confd** コマンドを実行します。

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

bulkstats

この NETCONF プロトコル コンフィギュレーション モード コマンドを使用すると、REST インターフェイスを介したバルク統計情報の収集とレポートが可能になります。デフォルトでは、このコマンドはディセーブルです。

コマンドシンタックスは **bulkstats** です。

REST インターフェイスを介して後から取得するため、StarOS の統計情報の収集中にバルク統計情報が CDB にも保存されます。

バルク統計情報の運用データを使用した ConfD の入力を無効にするには、**no bulkstats** を使用します。

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

confd-user

この NETCONF プロトコルのコンフィギュレーションモードのコマンドは、NETCONF を介してすべての CLI 操作のユーザー名を関連付けます。ユーザは、検証可能なログイン情報で認証されます。このユーザー名は、CLI ログイン目的でのみ使用されます。

コマンドシンタックスは、**confd-user <username>** です。ここで <username> は、1 ~ 144 文字の英数字の文字列です。



重要 NETCONF セッションまたは RESTful セッションは、検証可能なログイン情報で引き続き確立する必要があります。

kpi

この NETCONF プロトコルコンフィギュレーションモードでは、NSLB の重要業績評価指標 (KPI) の収集間隔を設定します。

コマンドシンタックスは **kpi seconds** であり、*seconds* は 0 (無効) または 10 ~ 120 の整数で、次の KPI を収集する時間間隔を秒単位で設定します。

- セッションによる CPU の使用率 (%)
- セッションによるメモリの使用率 (%)
- セッション以外による CPU の使用率 (%)
- セッション以外によるメモリ使用率 (%)
- セッションによる使用率 (%)

これらの統計情報は、システム全体で (すべてのカードで) キャプチャされます。デフォルトでは、この機能はディセーブルです。

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

netconf notifications events

この NETCONF プロトコル コンフィギュレーション モード コマンドは、StarOS に記録された イベントを「StarOS」という名前のストリームで NETCONF 通知として送信できるようにします。Level では、通知が生成される最も低いイベントシビラティ（重大度）レベルを指定します。

コマンドシンタックスは **netconf notifications events level { critical | error | warning | unusual | info }** で、レベルは次のようになります。

- **critical** - Level 1 : ログファイルに重大なエラーが含まれていることを報告します。
- **error** - Level 2 : ログファイルにエラー通知が含まれていることを報告します。
- **warning** - Level 3 : ログファイルに警告メッセージが含まれていることを報告します。
- **unusual** - Level 4 : ログファイルに予期しないエラーが含まれていることを報告します。
- **info** - Level 5 : ログファイルに情報メッセージが含まれていることを報告します。

NETCONF 通知を無効にするには **no netconf notifications events** を使用します。



重要 カテゴリ「critical-info」（シビラティ（重大度）を問わない）のイベントも通知に変換されません。

netconf notifications snmp

この NETCONF プロトコル コンフィギュレーション モード コマンドは、SNMP アラートとアラームに「StarOS_SNMP」という名前のストリームで NETCONF 通知として送信できるようにします。

コマンドシンタックスは **netconf notifications snmp** です。

NETCONF 通知を無効にするには **no netconf notifications snmp** を使用します。

netconf port

この NETCONF プロトコル コンフィギュレーション モード コマンドは、NETCONF インターフェイスのポート番号を設定します。**server confd** が有効になっている場合、デフォルトポートは自動的に 830 に設定されます。

コマンドシンタックスは **netconf port port_number** で、*port_number* は 1 ~ 65,535 の整数である必要があります。

ポート番号を 830 にリセットするには **no netconf port** を使用します。



重要 NETCONF インターフェイスのポート値を変更すると、ConfDの再起動が予定され、NETCONF および REST（有効になっている場合）インターフェイスを介した接続の一時的な切断が発生します。

rest auth-policy

この NETCONF プロトコル コンフィギュレーション モード コマンドは、サーバーがクライアント証明書に対して実行する検証のレベルを制御します。CA（認証局）証明書は、グローバル コンフィギュレーション モードで既存の **ca-certificate** コマンドを使用して設定できます。

コマンドシンタックスは **rest auth-policy { none | peer | peer-fail }** で、レベルは次のようになります。

- **none** : 認証は実行されません。
- **peer** : クライアントが証明書を提供しない場合、またはクライアントが証明書を提供し、それが有効である場合は、接続が許可されます。クライアントが無効な証明書を提供した場合、接続は中断されます。



重要 **peer** が選択されている場合は、CA 証明書が推奨されます。そうでない場合、有効な証明書を提供するクライアントは認証できず、接続は失敗します。

- **peer-fail** : サーバーはクライアントがクライアント証明書を提供するように要求し、証明書が正常に検証されない場合は接続に失敗します。



重要 **peer-fail** が選択されている場合は、1 つ以上の CA 証明書がデバイス上に存在する必要があります。そうでない場合は REST インターフェイスは有効になりません。

no rest auth-policy を使用して認証ポリシーを **none** に設定します。認証は実行されません。



重要 REST インターフェイスの認証ポリシーを変更すると、ConfDの再起動が予定され、NETCONF および REST（まだ有効になっている場合）インターフェイスを介した接続の一時的な切断が発生する可能性があります。

REST が有効になっている間に ConfD が使用しているグローバル証明書を変更すると、ConfD の再起動も発生させます。

rest certificate

この NETCONF プロトコル コンフィギュレーション モード コマンドは、REST インターフェイスの証明書と秘密キーを設定します。

コマンドシンタックスは **rest certificate** *certificate_name* で、*certificate_name* は 1 ～ 128 文字の英数字の文字列です。



重要 指定した証明書がデバイス上に存在している必要があります。証明書および関連付けられた秘密キーは、既存の **certificate** コマンドをグローバル コンフィギュレーション モードで使用して設定できます。

設定した証明書とキーを削除するには **no rest certificate** を使用します。有効な証明書とキーがないと REST は動作しません。



重要 REST インターフェイスの証明書を変更すると、ConfD の再起動が予定され、NETCONF および REST (有効になっている場合) インターフェイスを介した接続の一時的な切断が発生することがあります。

REST が有効になっている間に ConfD が使用しているグローバル証明書を変更すると、ConfD の再起動も発生させます。

rest hostname

この NETCONF プロトコル コンフィギュレーション モード コマンドで Web サーバーが対応するホスト名を指定します。設定されている場合、Web サーバーは、ホストフィールドが設定されたホスト名と一致する要求にのみ対応するよう求められます。

コマンドシンタックスは **rest hostname** *host_name* であり、*host_name* は 1 ～ 63 文字の英数字の文字列です。

システム名を使用するには **no rest hostname** を使用します。ホスト名との一致は求められません。



重要 REST インターフェイスのホスト名を変更すると、ConfD の再起動が予定され、NETCONF および REST (まだ有効になっている場合) インターフェイスを介した接続の一時的な切断が発生する可能性があります。

REST が有効になっている間に ConfD が使用しているグローバル証明書を変更すると、ConfD の再起動も発生させます。

rest port

この NETCONF プロトコル コンフィギュレーション モード コマンドは、REST インターフェイスのポート番号を設定します。

コマンドシンタックスは **rest port port_number** で、*port_number* は 1 ~ 65,535 の整数である必要があります。

ポート番号をデフォルトの 443 にリセットするには **no rest port** を使用します。



重要 REST インターフェイスのポート値を変更すると、ConfD の再起動が予定され、NETCONF および REST (まだ有効になっている場合) インターフェイスを介した接続の一時的な切断が発生する可能性があります。

REST が有効になっている間に ConfD が使用しているグローバル証明書を変更すると、ConfD の再起動も発生させます。

設定例

次のコマンドシーケンスは、NETCONF プロトコルをサポートする ConfD 設定を確立します。

server confd を有効にするには、タイプ v2-RSA SSH キーが必要です。

```
configure
  context local
    ssh key
  <encrypted key text>
  len 938 type v2-rsa
  server confd
    bulkstats
    confd-user NETCONF
    rest certificate rest-cert
  #exit
  subscriber default
  exit
  aaa group default
  #exit
  gtpm group default
  #exit
#exit
end
```

注：

- **bulkstats**、**confd-user**、および **rest** はオプションです。**server confd** を設定すると、NETCONF サポートが有効になります。

設定の確認

NETCONF-ConfD 設定に関する情報を表示するには、2 つの Exec モードの **show** コマンドがあります。

show confdmgr コマンド

このコマンドは、StarOS ConfD Manager (confdmgr) プロセスに関する情報を表示します。

このコマンドの構文は、次のとおりです。

```
show confdmgr [ confd { cdb | netconf | state } | model { bulkstats |
confd } | subscriptions ] [ | { grep grep_options | more } ]
```

注：

- **confd** キーワードは、次のオプションの指定されたキーワードに基づいて ConfD エンジンに関する情報を表示します。
 - **cdb** は ConfD CDB 情報を表示します。
 - **netconf** は NETCONF ステート情報を表示します。
 - **state** は現在の ConfD ステート情報を表示します。
- **model** キーワードは、次のオプションの指定されたキーワードに基づいて ConfD モデルに関する情報を表示します。
 - **bulkstats** : バルク統計情報の設定と運用データ
 - **confd** : サーバー ConfD 設定
- サブスクリプションキーワードを指定すると、ConfD CDB サブスクリプション情報が表示されます。

show confdmgr

show confdmgr の出力例については、以下を参照してください。

```
[local]<host_name># show confdmgr

State Information
-----
State                Started
Subscriptions        5
Last successful id   1461-704882-705350
Last failed id      None
Username             Not configured
Bulkstats            Enabled
Kpi interval         30
Event notification level Disabled
SNMP notifications  Disabled
REST interface authentication none
REST interface certificate rest-cert
REST interface host name Not configured

Interface            Status            Port
-----
NETCONF              Enabled           830
REST                 Enabled           443

Statistics
-----
```

```

Triggers                1
Replays                 0
Notifications           5
Notification failures   0
Trigger failures        0
Replay failures         0
NETCONF notification failures 0
Unexpected failures     0
[local]<host_name>#

```

この出力の統計部分には、次の情報が含まれています。

- **Triggers** : confdmgr が CDB の内容をダンプして confdmgr に戻すように ConfD に要求した結果、SCT への設定の同期が行われた回数（共有設定タスク）。
- **Replays** : トランザクションが再生された回数。再生は、起動時に confdmgr で最後に成功したトランザクション ID が ConfD と一致しない場合に開始されます。たとえば、これは設定トランザクションの通知を処理するとき confdmgr タスクが再起動された場合に発生する可能性があります。
- **Notifications** : ConfD が confdmgr に設定の更新を送信した回数。たとえば、これは confd_cli を介した「commit」か、またはトリガーイベントの結果として発生することがあります。
- **Failed Notifications** : ConfD から受信した設定が正常に処理されなかった回数。
- **Trigger failures** : confdmgr への CDB ダンプが失敗した回数。
- **Replay failures** : トランザクションの再生試行が失敗した回数。
- **NETCONF notification failures** : NETCONF 通知の発行試行が失敗した回数。
- **Unexpected failures** : 予期しない状況が発生した回数。各ケースに対してエラーログが生成されます。

show confdmgr confd cdb

show confdmgr confd cdb の出力例については、次を参照してください。

```

[local]<host_name># show confdmgr confd cdb
bulkstats server collection true
bulkstats server historical-collection false
bulkstats server gather-on-standby true
bulkstats server sample-interval 60
bulkstats server transfer-interval 1440
bulkstats server limit 7500
bulkstats server receiver-mode secondary-on-failure
bulkstats server file 1
!
bulkstats schemas file 1
schema-type system
  schema abc
  format      %host%
  active-only false
!
schema common
  format      %host%,%ipaddr%,%time%,%uptime%,%swbuild%,%localtz%
  active-only false
!
schema systemSch11

```

```

format
PPM,system,systemSch11,%epochtime%,%localdate%,%localtime%,%uptime%,%diameter-msg
-saans%,%diameter-msg-sarretry%,%diameter-msg-saatimeout%,%diameter-msg-saadropped%,%diameter-ms
g-uareq%,%diameter-msg-uaans%,%diameter-msg-uarretry%,%diameter-msg-uaatimeout%,%diameter-msg-ua
adropped%,%diameter-msg-lireq%,%diameter-msg-lians%,%diameter-msg-lirretry%,%diameter-msg-liatim
eout%,%diameter-msg-liadropped%,%diameter-msg-rtreq%,%diameter-msg-rtans%,%diameter-msg-rtrejec
t%,%diameter-msg-ppreq%,%diameter-msg-ppans%,%diameter-msg-ppreject%,%diameter-msg-dereq%

    active-only false
    !
    !
    !
confd bulkstats true
confd netconf port 830
confd rest port 443
confd rest auth-policy none
confd rest certificate rest-cert
nacm read-default permit
nacm groups group admin
!
nacm groups group inspector
!
nacm groups group operator
!
nacm groups group secure_admin
!
nacm rule-list secure_admin
!
group [ secure_admin ]
rule any-access
    action permit
!
rule secure_admin_server_confid
module-name      cisco-staros-cli-config
path             /context/server/confd
access-operations create,read,update
action           permit
|
|
V
nacm rule-list inspector
group [ inspector ]
rule any-access
    access-operations read
    action             permit
!
!
[local]<host_name>#

```

show confdmgr confd netconf

show confdmgr confd netconf の出力例については、以下を参照してください。

```

[local]<host_name># show confdmgr confd netconf
netconf-state capabilities capability urn:ietf:params:netconf:base:1.0
netconf-state capabilities capability urn:ietf:params:netconf:base:1.1
netconf-state capabilities capability
urn:ietf:params:netconf:capability:writable-running:1.0
netconf-state capabilities capability urn:ietf:params:netconf:capability:candidate:1.0
|
|
V
netconf-state statistics netconf-start-time 2016-03-30T17:09:49-04:00
netconf-state statistics in-bad-hellos 0

```

```

netconf-state statistics in-sessions 0
netconf-state statistics dropped-sessions 0
netconf-state statistics in-rpcs 0
|
|
V
netconf-state datastores datastore candidate
netconf-state schemas schema cisco-staros-bulkstats 2016-12-14 yang
  namespace http://www.cisco.com/staros-bulkstats
  location [ NETCONF ]
netconf-state schemas schema cisco-staros-bulkstats-config 2016-12-14 yang
  namespace http://www.cisco.com/staros-config
  location [ NETCONF ]
|
|
V
NAME                                CREATOR  CREATED                                CONTEXT
-----
/rollback0                          system   2017-01-17T13:40:53-00:00             system
/rollback1                          system   2017-01-17T13:40:52-00:00             system
/rollback2                          system   2017-01-17T13:40:52-00:00             system
/rollback3                          system   2017-01-17T13:40:52-00:00             system
/rollback4                          system   2017-01-17T13:36:43-00:00             system
|
|
V
/cli-history/admin.hist
/cli-history/root.hist
/global.data

netconf-state streams stream NETCONF
  description "default NETCONF event stream"
  replay-support false
netconf-state streams stream StarOS
  description "StarOS Notifications"
  replay-support true
  replay-log-creation-time 2017-02-10T16:00:59+00:00
[local]<host_name>#

```

show confdmgr confd state

show confdmgr confd state の出力例については、以下を参照してください。

```

[local]<host_name># show confdmgr confd state
Monday June 24 10:58:49 EDT 2019
confd-state version 7.1
confd-state epoll false
confd-state daemon-status started
confd-state loaded-data-models data-model acs-config
  revision 2016-10-31
  namespace http://www.cisco.com/usp/nfv/acs-config
  prefix acs-config
  exported-to-all
confd-state loaded-data-models data-model cisco-staros-bulkstats
  revision 2016-12-14
  namespace http://www.cisco.com/staros-bulkstats
  prefix staros_bulkstats
  exported-to-all
confd-state loaded-data-models data-model cisco-staros-cli-config
  revision 2016-12-14
  namespace http://www.cisco.com/staros-cli-config
  prefix staros_cli
  exported-to-all
confd-state loaded-data-models data-model cisco-staros-config

```

```

revision          2016-12-14
namespace         http://www.cisco.com/staros-config
prefix           staros_config
exported-to-all
confd-state loaded-data-models data-model cisco-staros-exec
revision          2016-12-14
namespace         http://www.cisco.com/staros-exec
prefix           staros_exec
exported-to-all
confd-state loaded-data-models data-model cisco-staros-kpi
revision          2017-10-31
namespace         http://www.cisco.com/staros-kpi

```

show confdmgr model bulkstats

show confdmgr model bulkstats の出力例については、次を参照してください。

```
[local]<host_name># show confdmgr model bulkstats
```

```

Model: Bulkstats
-----

Operational Data:
  Requests          277
  Records           831
  Failures           0

Configuration:
  CLI updates       0
  NETCONF updates   2
  Aborts            0
  Failures          0
local]<host_name>#

```

この出力の運用データ部分には、次の情報が含まれています。

- **Requests** : bulkstats から confdmgr への運用データメッセージ要求の数。
- **Records** : 処理された運用データスキーマレコードの数。
- **Failures** : バルク統計情報からのプッシュ要求の処理中に confdmgr 内で検出されたエラーの数。

この出力の設定部分には、次の情報が含まれています。

- **CLI updates** : CLI からのプッシュ設定要求の数と、SCT からの設定のロードの回数。
- **NETCONF updates** : バルク統計情報のサブスクリプション通知の数。
- **Aborts** : NETCONF を介した設定更新が中断された回数。
- **Failures** : confdmgr 内のバルク統計情報設定要求の処理で検出されたエラーの数。

show confdmgr model confd

show confdmgr model confd の出力例については、次を参照してください。

```
[local]<host_name># show confdmgr model confd
```

```
Model: ConfD
```

```

-----
CLI updates      0
NETCONF updates  1
Aborts           0
Failures         0
local]<host_name>#

```

- **CLI updates** : CLI からのプッシュ設定要求の数と、SCT からの設定のロードの回数。
- **NETCONF updates** : ConfD 設定サブスクリプション通知の数。
- **Failures** : confdmgr 内の ConfD 設定要求の処理で検出されたエラーの数。
- **Aborts** : NETCONF を介した設定更新が中断された回数。

show confdmgr subscriptions

show confdmgr subscriptions の出力例については、以下を参照してください。

```

[local]<host_name># show confdmgr subscriptions

Subscriptions:
Path                               Index  Namespace
-----
/active-charging                    6      http://www.cisco.com/staros-cli-con
fig
/context                             7      http://www.cisco.com/staros-cli-con
fig
/bulkstats/server                   8      http://www.cisco.com/staros-config
/bulkstats/schemas                 9      http://www.cisco.com/staros-config
/confd                              10     http://www.cisco.com/staros-config
[local<host_name>#

```

サブスクリプションとは、変更が発生したときに **confdmgr** に通知する必要がある Yang モデルで定義されている設定ポイントのことです。

clear confdmgr confd cdb

この Exec モードコマンドは、ConfD が設定オブジェクトを保存するために使用する ConfD 設定データベース (CDB) の設定を消去します。StarOS は、ConfD が提供する API を使用してデータベースにアクセスします。



- (注) ConfD および NETCONF プロトコルのサポートを無効にするために、コンテキスト コンフィギュレーション モードの **no server confd** コマンドがローカルコンテキストで実行されている場合を除き、CDB を消去することはできません。

次に、CDB をクリアするためのコマンドシーケンスの例を示します。

```

[local]host_name# config
[local]host_name(config)# context local
[local]host_name(config-ctx)# no server confd
[local]host_name(config-ctx)# end
[local]host_name# clear confdmgr confd cdb
About to delete the ConfD configuration database

```



```
The running configuration is NOT affected.  
Are you sure? [Yes|No]: Y  
[local]host_name#
```



注意 CDB のクリアは端末の動作です。CDB はコンテキスト コンフィギュレーション モードの **server confd** コマンドがローカルコンテキストで実行されているときに再度追加されて、ConfD と NETCONF プロトコルサポートを再度有効にします。

clear confdmgr statistics

このコマンドは、**show confdmgr** コマンドの出力の [Statistics] セクションに掲載されているすべての内容がクリアされます。これには、次が含まれます。

- Triggers
- Replays
- Notifications
- Notification failures
- Trigger failures
- Replay failures
- NETCONF notification failures
- Unexpected failures

YANG モデル

StarOS のインストールでは、次の YANG ファイルを使用できます。

- **cisco-staros-bulkstats-config.yang** : StarOS のネイティブバルク統計設定モデル。
- **cisco-staros-bulkstats-schema-types.yang** : **cisco-staros-bulkstats-config.yang** モデルの拡張機能。コードから直接取得したスキーマ名を列挙したリストが含まれています。
- **cisco-staros-bulkstats.yang** : RESTful インターフェイスを介して顧客がバルク統計を取得できるようにする運用データモデル。このモデルは、管理者クレデンシャルを持つユーザーのみが使用できます。
- **cisco-staros-confd-config.yang** : ネイティブのサーバー ConfD 設定モデル。
- **cisco-staros-config.yang** : 他のすべての **cisco-staros-*** 設定モデルを含めるために使用されるコンテナ YANG ファイル (すべてのネイティブモデルが共通名前空間下のこの場所に含まれています)。

- **cisco-staros-exec.yang** : RESTful インターフェイスを介した CLI の実行動作を可能にするモデル。このモデルは、管理者クレデンシャルを持つユーザーのみが使用できます。入力を解析するために ConfD によってローカルに使用されます。
- **cisco-staros-notif.yang** : StarOS のイベントロギングの NETCONF 通知ストリームを可能にするモデル。デバッグレベルのイベントはサポートされていません。情報メッセージ以上のみがサポートされています。



重要 これらの YANG ファイルが読み込まれて使用可能になる前に、ConfD サーバーを少なくとも 1 回起動する必要があります。

StarOS ネットワーク要素ドライバ (NED) を構築するには、YANG ファイルを Cisco NSO にプルする必要があります。

YANG ファイルをコピーするには、次のようなコマンドを入力します。

```
#copy /hd-raid/confd_dir/etc/confd/cisco-staros-confd-config.yang
sftp://<user>:<password>@<host>/sftp-directory/cisco-staros-confd-config.yang
```

Show Support Details (SSD)

すべての **show confdmgr** コマンドの出力が SSD に追加されました。

ConfD の例

サーバー ConfD

次に、完全な TLS 認証と curl を使用してサーバー ConfD 設定を取得する例を示します。

サーバー ConfD の設定

次に、デフォルト以外の NETCONF と HTTPS ポートを使用して RESTful インターフェイスが有効になっているサーバー ConfD の設定例を示します。

```
[local]<host_name># show configuration confd
[local]<host_name># config
[local]<host_name>(config)# ca-certificate name ca-cert pem url /flash/ssl/rootCA.pem
[local]<host_name>(config)# certificate name rest-cert pem url /flash/ssl/host.crt
private-key pem url /flash/ssl/host.key
[local]<host_name>(config)# end
[local]<host_name># config
[local]<host_name>(config)# context local
[local]<host_name>(config-ctx)# server confd
[local]<host_name>(config-confd)# netconf port 123
[local]<host_name>(config-confd)# rest port 234
[local]<host_name>(config-confd)# rest certificate rest-cert
[local]<host_name>(config-confd)# rest auth-policy peer-fail
```

```
[local]<host_name>(config-confd)# end
[local]<host_name># show confdmgr

State Information
-----
State                               Started
Subscriptions                        5
Last successful id                   1488-211047-99241
Last failed id                       None
Username                             Not configured
Bulkstats                            Disabled
Event notification level             Disabled
SNMP notifications                  Disabled
REST interface authentication        peer-fail
REST interface certificate            rest-cert
REST interface host name             Not configured

Interface          Status      Port
-----
NETCONF            Enabled    123
REST               Enabled    234

Statistics
-----
Triggers           1
Replays            0
Notifications      8
Notification failures 0
Trigger failures   0
Replay failures    0
NETCONF notification failures 0
Unexpected failures 0
```

netconf-console を使用したサーバー ConfD 設定の取得

次に、NETCONF を介してサーバー ConfD 設定を取得するための netconf-console の使用例を示します。

```
[user@server]$ ./netconf-console --host 1.2.3.4 -u admin --password pswd! --port 123
--get-config -x confd
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="1">
  <data>
    <confd xmlns="http://www.cisco.com/staros-config">
      <bulkstats>>false</bulkstats>
      <netconf>
        <port>123</port>
      </netconf>
      <rest>
        <port>234</port>
        <auth-policy>peer-fail</auth-policy>
        <certificate>rest-cert</certificate>
      </rest>
    </confd>
  </data>
</rpc-reply>
```

注：

- netconf-console は GitHub (<https://github.com/tail-f-systems/JNC/blob/master/examples/2-junos/netconf-console>) から自由に使用できます。

Curl を使用したサーバー ConfD 設定の取得

次に、curl を使用して同じ **get-config** 操作を実行する例を示します。

```
[<user>@server] ]$ curl -u admin:pswd!
https://rtp-mitg-si06.cisco.com:234/api/running/confd?deep --cert
/users/<user>/ssl_cert/client_cert/client.crt --key
/users/<user>/ssl_cert/client_cert/client.key --cacert
/users/<user>/ssl_cert/root_cert/rootCA.pem

<confd xmlns="http://www.cisco.com/staros-config" xmlns:y="http://tail-f.com/ns/rest"
xmlns:staros_config="http://www.cisco.com/staros-config">
  <bulkstats>false</bulkstats>
  <netconf>
    <port>123</port>
  </netconf>
  <rest>
    <port>234</port>
    <auth-policy>peer-fail</auth-policy>
    <certificate>rest-cert</certificate>
  </rest>
</confd>
```

バルク統計情報

次に、バルク統計情報の運用データの例を示します。

バルク統計情報の有効化

サーバー ConfD で **bulkstats** を有効にします。

```
[local]<host_name># config
[local]<host_name>(config)# context local
[local]<host_name>(config-ctx)# server confd
[local]<host_name>(config-confd)# bulkstats
[local]<host_name>(config-confd)# end
[local]<host_name># show confdmgr

State Information
-----
State                Started
Subscriptions        5
Last successful id   1488-216669-170664
Last failed id       None
Username             Not configured
Bulkstats            Enabled
Event notification level Disabled
SNMP notifications  Disabled
REST interface authentication peer-fail
REST interface certificate rest-cert
REST interface host name Not configured

Interface            Status          Port
-----
NETCONF             Enabled         123
REST                Enabled         234
```

バルク統計情報の設定

バルク統計情報の設定例については、次を参照してください。

```
[local]<host_name># show config bulkstats
config
  bulkstats collection
  bulkstats mode
    file 1
      schema common format %uptime%,%host%,%ipaddr%
    #exit
    file 2
      schema system format %uptime%,%host%,%ipaddr%
    #exit
  #exit
End
```

バルク統計情報収集の強制

詳細情報を強制的に収集して、ConfDの運用データベースにプッシュする例については、次を参照してください。

```
[local]<host_name># bulkstats force gather
```

注：

- 統計情報は一般に、bulkstats に設定された収集間隔タイマーごとにプッシュされます。

Curl を使用した統計情報の読み取り

サーバー ConfD RESTful インターフェイスを介して統計情報を読み取るための Curl の使用例については、次を参照してください。

```
[<user>@server] ]$ curl -u admin:pswd!
https://rtp-mitg-si06.cisco.com:234/api/operational/bulkstats-operational?deep --cert
/users/<user>/ssl_cert/client_cert/client.crt --key
/users/<user>/ssl_cert/client_cert/client.key --cacert
/users/<user>/ssl_cert/root_cert/rootCA.pem
```

```
<bulkstats-operational xmlns="http://www.cisco.com/staros-bulkstats"
xmlns:y="http://tail-f.com/ns/rest"
xmlns:staros_bulkstats="http://www.cisco.com/staros-bulkstats">
  <file>
    <number>1</number>
    <schemas>
      <schema>system</schema>
      <names>
        <name>common</name>
        <key_ids>
          <key_id>none</key_id>
          <variable>
            <name>host</name>
            <value><host_name></value>
          </variable>
          <variable>
            <name>ipaddr</name>
            <value>1.2.3.4</value>
          </variable>
          <variable>
            <name>uptime</name>
            <value>5781</value>
          </variable>
        </key_ids>
      </names>
    </schemas>
```

```

</file>
<file>
  <number>2</number>
  <schemas>
    <schema>system</schema>
    <names>
      <name>system</name>
      <key_ids>
        <key_id>none</key_id>
        <variable>
          <name>host</name>
          <value><host_name></value>
        </variable>
        <variable>
          <name>ipaddr</name>
          <value>1.2.3.4</value>
        </variable>
        <variable>
          <name>uptime</name>
          <value>5781</value>
        </variable>
      </key_ids>
    </names>
  </schemas>
</file>
</bulkstats-operational>

```

Exec CLI モデル

次の例では、Exec CLI モデルを使用しています。

Curl を使用した「show version」出力の取得

Curl を使用して **show version** の出力を取得する例については、次を参照してください。

```

cat exec_cli_show_version.xml
<input><args>show version</args></input>
*****
[<user>@server] ]$ curl -u admin:pswd!
https://rtp-mitg-si06.cisco.com:234/api/running/staros_exec/_operations/exec --cert
/users/<user>/ssl_cert/client_cert/client.crt --key
/users/<user>/ssl_cert/client_cert/client.key --cacert
/users/<user>/ssl_cert/root_cert/rootCA.pem -X POST -T ./exec_cli_show_version.xml
<output xmlns='http://www.cisco.com/staros-exec'>
  <result>Active Software:
    Image Version:      21.2.M0.private
    Image Build Number: private
    Image Description:  Developer_Build
    Image Date:         Thu Feb 23 15:25:47 EST 2017
    Boot Image:         /flash/qvpc-si.bin.conf
    Source Commit ID:   bd234043a93c68873ea77444733a8c632356d161
  </result>
</output>

```

Curl を使用した複数の show コマンド出力の取得

次に、コマンド間のデリミタとして「\r\n」を使用した **show build** および **show confdmgr** の出力を取得するための curl の使用例を示します。

```

cat exec_cli_show_build_and_confdmgr.xml
<input><args>show build \r\n show confdmgr</args></input>
*****
[<user>@server] ]$ curl -u admin:pswd!
https://rtp-mitg-si06.cisco.com:234/api/running/staros_exec/_operations/exec --cert
/users/<user>/ssl_cert/client_cert/client.crt --key
/users/<user>/ssl_cert/client_cert/client.key --cacert
/users/<user>/ssl_cert/root_cert/rootCA.pem -X POST -T ./
exec_cli_show_build_and_confdmgr.xml
<output xmlns='http://www.cisco.com/staros-exec'>
  <result>Active Software:
    Image Version:                21.2.M0.private
    Image Build Number:           private
    Image Description:            Developer_Build
    Image Date:                   Thu Feb 23 15:25:47 EST 2017
    Boot Image:                   /flash/qvpc-si.bin.confd
    Source Commit ID:             bd234043a93c68873ea77444733a8c632356d161
    Kernel Version:               2.6.38-staros-v3-ssi-64
    Kernel Machine Type:          x86_64

Build Information:
  Kernel Build:                   #1 SMP PREEMPT Wed Feb 22 12:28:49 EST 2017
  Image Build Type:               Production build
  Image Build User:               <user>
  Image Build Machine:            <host_name>
  Image Build Changeset Version:  +
  Image Build Changeset Author:   <user>
  Image Build Changeset Location: cisco.com
  Image Build Changeset Number:   bd234043a93c68873ea77444733a8c632356d161
  Image Build Changeset PID:      2017-02-23
*****
**** Local changes exist ****
*****

State Information
-----
State                Started
Subscriptions        5
Last successful id   1488-216669-170664
Last failed id      None
Username             Not configured
Bulkstats            Enabled
Event notification level Disabled
SNMP notifications  Disabled
REST interface authentication peer-fail
REST interface certificate rest-cert
REST interface host name Not configured

Interface            Status            Port
-----
NETCONF              Enabled           123
REST                 Enabled           234

Statistics
-----
Triggers              1
Replays               0
Notifications        27
Notification failures 0
Trigger failures     0
Replay failures       0
NETCONF notification failures 0
Unexpected failures   0
</result>

```

```
</output>
*****
```

ConfD のアップグレードサポート

ECS コマンド用の CLI ベースの YANG モデル

このリリースでは、**cisco-staros-cli-config.yang** モデルは、NSO を介して限定された一連の ECS (Enhanced Charging System) 設定コマンドをサポートします。

サウスバウンド側では、ConfD は、ConfD 管理エージェントによって提供される一連の API を介して呼び出された StarOS プロセスと通信します。ConfD CDB は、オブジェクトを保存するために ConfD によって使用されます。StarOS は、ConfD 提供の API を使用してデータベースにアクセスします。ConfD コンフィギュレーション データベースにデータが入力されると、StarOS は CLI アクセスによる全体的な設定の変更を引き続き許可します。結果として、CDB への自動更新は行われません。CDB は NETCONF インターフェイス経由でアップデートを受信します。CDB と StarOS コンフィギュレーション データベースを同期された状態に保つには、**cisco-staros-cli-config** YANG モデルでサポートされる設定オブジェクトに加えられた CLI アクセス (外部から NETCONF) を介したすべての変更を手動で CDB に適用する必要があります。

CDB のシーディングと同期

server confd を有効にした後、StarOS で ConfD によって管理されているコンフィギュレーション データベース (CDB) のローカルコピーを使用して CDB を最初にシードする必要がある場合があります。シーディングの手順では、CLI ベースのすべての YANG モデルでサポートされている設定コマンドを含む StarOS プラットフォームで ConfD によって使用される CDB が作成されます。



- 重要**
- StarOS CLI を介して管理対象オブジェクトを手動で変更した場合は、次に説明する手順を繰り返して、実行時設定を NSO と再同期する必要があります。

- ステップ 1** Exec モードの **save configuration <url> confd** を実行して、ConfD でサポートされている StarOS 設定データを /flash デバイス上のファイルに保存します。
- ステップ 2** Exec モードの **show configuration error** を実行して、保存された設定を検証します。設定を適用する前にエラーを修正します。そうしないと、ConfD によって設定全体が拒否されます。
- ステップ 3** Exec モードの **configure confd <url>** を実行して、ConfD 設定を適用します。ConfD 設定が適用されると、デバイスは NSO 管理サービスへの NETCONF 接続を確立できるようになります。

ステップ 4 デバイスを NSO と同期します。同期プロセスの詳細については、NSO のユーザーマニュアルを参照してください。

show configuration confd コマンド

confd キーワードは **show configuration** コマンドの出力をフィルタリングして、CLI ベースの YANG モデルでサポートされている設定コマンドだけを表示します。

```
show configuration confd
```

サンプル出力は次のようになります。

```
[local]<host_name># show configuration confd
config
  context local
    server confd
    #exit
  active-charging service ecs
    ruledef rdl
      tcp any-match = TRUE
    #exit
  rulebase default
    #exit
#exit
end
[local]<host_name>#
```

CDB のメンテナンス

ConfD コンフィギュレーションデータベース (CDB) のローカルコピーは、StarOS 上の ConfD によって管理されます。

すべての ConfD でサポートされている StarOS コンフィギュレーション コマンドを URL に表示して保存できます。そのため、**confd** キーワードが **show configuration** コマンドと **save configuration** コマンドに追加されました。

ConfD でサポートされている設定を URL に保存した後、Exec モードの **configure confd <url>** コマンドを使用して、それを CDB に直接適用できます。このコマンドは、*url* にあるファイルの内容を、ConfD の実行コンフィギュレーションに適用します。

上記のコマンドに関する詳細については、以下を参照してください。

clear confdmgr confd cdb

この Exec モードコマンドは、ConfD が設定オブジェクトを保存するために使用する ConfD 設定データベース (CDB) の設定を消去します。StarOS は、ConfD が提供する API を使用してデータベースにアクセスします。



- (注) ConfD および NETCONF プロトコルのサポートを無効にするために、コンテキスト コンフィギュレーション モードの **no server confd** コマンドがローカルコンテキストで実行されている場合を除き、CDB を消去することはできません。

次に、CDB をクリアするためのコマンドシーケンスの例を示します。

```
[local]host_name# config
[local]host_name(config)# context local
[local]host_name(config-ctx)# no server confd
[local]host_name(config-ctx)# end
[local]host_name# clear confdmgr confd cdb
About to delete the ConfD configuration database
The running configuration is NOT affected.
Are you sure? [Yes|No]: y
[local]host_name#
```



注意 CDB のクリアは端末の動作です。その後、CDB を再設定する必要があります。

configure confd <url>

この Exec モードのコマンドは、URL によって指定された設定スクリプトの内容を現在の ConfD 設定データベース (CDB) に適用します。

次に、コマンドシーケンスの例を示します。

```
[local]host_name# save configuration /flash/confd.config confd
[local]host_name# configure confd /flash/confd.config
Info: #!$$ StarOS V20.2 Chassis 52767e9ff9e207bed12c76f7f8a5352c
Info: config
Info:   active-charging service acs
Info:   rulebase default
Info:   #exit
Info:   #exit
Info: end
[local]host_name#
```

save configuration <url> confd

キーワード **confd** が Exec モードの **save configuration** コマンドに追加されました。このキーワードは、YANG モデルでサポートされている設定コマンドのみを含むように、保存された設定コマンドをフィルタリングします。

次に、このプロセスのコマンドシンタックスを示します。

```
[local]host_name# save configuration <url> confd
```

設定コマンドの YANG モデルのサブセット出力は、**show file url <url>** コマンドを使用して表示できます。ここで、<url>は、設定を保存するために使用されるパス名です。保存された設定ファイルは、**configure confd** コマンドを使用して CDB に適用できます。

サポートされている StarOS ECS 設定コマンド

このリリースでは、CLI ベースの YANG モデルで次の StarOS ECS コマンドがサポートされています。

- **ruledef <ruledef_name>**
 - ip server-ip-address = *
 - tcp-ether-port = *
 - udp ether-port = *
 - tcp ether-port-range = *
 - udp ether-port range = *
 - tcp-any-match = *
 - udp any-match = *
 - http url = *
 - httpcookie = *
 - http x-header = *
- **group-of-ruledefs <ruledefs_group_name>**
 - add-ruledef priority = *
- **qos-group-of-ruledefs <group_name>**
 - add-group-of-ruledef <group_of_ruledef_name>
- **charging-action <charging_action_name>**
 - flow-idle-timeout <seconds>
 - content-id l
 - service-identifier <service_id>
 - billing-action egcdr
- **rulebase <rulebase_name>**
 - action priority <priority_number> group-of-ruledefs <ruledefs_group_name> charging-action <charging_action_name>



(注) "=" indicates support for every option following the prior keyword/value.

翻訳について

このドキュメントは、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。