



Cisco Unified Border Element (SP Edition) SPA DSP サービス

共有ポートアダプタ (SPA) のデジタルシグナルプロセッサ (DSP) は、複数のシスコプラットフォームで使用できるシングル幅、ハーフハイト、高出力の SPA モジュールです。SPA DSP は、シスコのミッドレンジおよびハイエンドルータ上の SPA の、DSP ベースの音声およびビデオソリューション用に設計されています。

Cisco IOS XE Release 3.2S で、次の SPA DSP 機能が、セッションボーダーコントローラ (SBC) 用に Cisco ASR 1000 シリーズルータに導入されました。

- DSP ファームプロファイルへの SBC の設定の関連付け。
- オンボード DSP サービスを使用した音声トランスコーディングとトランスレーティングのサポート。
- オンボード DSP サービスを使用したデュアルトーン多重周波数 (DTMF) インターワーキング。
- VoIPv4 および VoIPv6 のトランスコーディングとトランスレーティングのサポート。
- トランスコーディング、トランスレーティング、および DTMF インターワーキングのコール制御とシグナリング制御。

Cisco Unified Border Element (SP Edition) は、以前は Integrated Session Border Controller と呼ばれており、このマニュアルでは SBC と呼びます。

本章で使用されているコマンドの詳細な説明については、『*Cisco Unified Border Element (SP Edition) Command Reference: Unified Model*』

(http://www.cisco.com/en/US/docs/ios/sbc/command/reference/sbcu_book.html) を参照してください。

すべての Cisco IOS コマンドの詳細については、<http://tools.cisco.com/Support/CLILookup> にある Command Lookup Tool を使用するか、Cisco IOS マスター コマンドリストを参照してください。

Cisco Unified Border Element (SP Edition) の SPA DSP の機能履歴

リリース	変更内容
Cisco IOS XE Release 3.2S	SPA DSP オンボード サービスが、Cisco ASR 1000 シリーズルータに追加されました。
Cisco IOS XE Release 3.3S	コールリカバリ機能が追加されました。

内容

- 「制約事項」 (P.884)
- 「SPA DSP サービスの前提条件」 (P.884)
- 「SPA DSP サービスについて」 (P.884)
- 「SBC の SPA DSP サービスの設定」 (P.888)
- 「DBE 上の DSP ファーム プロファイルの設定」 (P.892)
- 「SDP の分析を非アクティブにするように SBC を設定」 (P.893)
- 「Unified SBC の設定」 (P.896)
- 「SBC の SPA DSP サービスの設定例」 (P.919)
- 「DBE の DSP ファーム プロファイルの設定例」 (P.920)
- 「SBC の SPA DSP サービスの設定例」 (P.919)
- 「Unified SBC の設定例」 (P.920)

制約事項

SPA DSP では、次の制限事項が適用されます

- 音声、オーディオ、およびビデオ会議はサポートされません。
- HA、システム レベルのインサーブिस ソフトウェア アップグレード (ISSU)、およびノンストップ フォワーディング (NSF) はサポートされません。
- ビデオ コーデックはサポートされません。
- 活性挿抜 (OIR) はサポートされますが、取り外し時に SPA を通過するセッションが失われます。
- Cisco Unified Communications Manager はサポートされません。

SPA DSP サービスの前提条件

DSP ファームの定義と SBC の設定およびアクティブ化は、SBC コールをトランスコードする前に実行する必要があります。SPA の設定の詳細については、次の場所にある『*Cisco ASR 1000 Series Aggregation Services Routers SIP and SPA Software Configuration Guide*』の「Configuring the Cisco DSP SPA for the ASR 1000 Series」を参照してください。

http://www.cisco.com/en/US/docs/interfaces_modules/shared_port_adapters/configuration/ASR1000/asrcfgdsp.html

SPA DSP サービスについて

SPA DSP には、SBC 用の音声トランスコーディング機能を提供するための、デジタル信号プロセッサと関連するハードウェアが含まれます。また、Cisco Unified Border Element, Enterprise は、単純な音声トランスコーディング サービス用に SPA DSP を使用できます。

SBC のトランスコーディング

SBC トランスコーディングは、Data Border Element (DBE) 機能の一部として 2 つの VoIP ネットワーク間のコーデック変換に使用されます。図 1 に、SPA DSP が Unified SBC のコーデック トランスコーディングを実行する方法を示し、図 2 に、SPA DSP が分散 SBC のコーデック トランスコーディングを実行する方法を示します。

図 1 Unified SBC の SPA DSP トランスコーディング

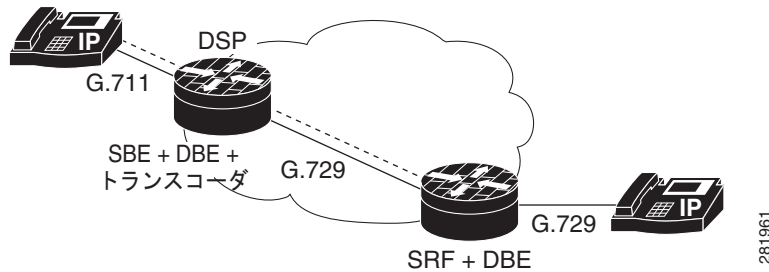
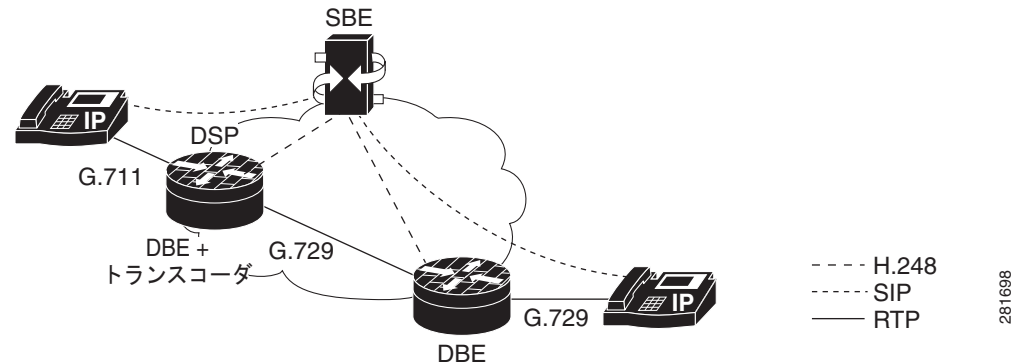


図 2 分散 SBC の SPA DSP トランスコーディング



SPA DSP は、ある種類のメディア ストリームまたはコーデックから、異なるメディア符号化と復号化テクノロジーを使用した別の種類のメディア ストリームまたはコーデックへの変換を可能にします。その他の変換アクティビティは次のとおりです。

- 異なるコーデックの間の変換
- 異なるパケット化設定間の変換（トランスレーティング）
- DTMF インターワーキング

分散 SBC のトランスコーディング

トランスコーディングは、コールをプログラムするために使用されるセッション記述プロトコル (SDP) から推定されます。異なるコーデックを含む同じコールの終端のプログラミングは、トランスコーディングの実行を分散 SBC に暗黙的に指示します。

分散 SBC のトランスレーティング

トランスレーティングはコールをプログラムするために使用される SDP から推定されます。ptime が異なる同じコールの終端のプログラミングは、トランスレーティングの実行を分散 SBC に暗黙的に指示します。



(注)

トランスレーティングは、コーデック間ではなく、同じコーデックを使用した異なるレートに対してのみサポートされます。したがって、トランスレーティングとトランスコーディングは同時に実行できません。

インバンド DTMF インターワーキング

Cisco ASR 1000 シリーズ アグリゲーション サービス ルータは、Real-Time Transport Protocol (RTP) インバンド ウェーブフォーム、RTP telephone-event コーデック (RFC2833)、および SIP DTMF インジケーション タイプ間の DTMF インターワーキングをサポートしています。

DTMF トーンは、次の方法を使用して生成できます。

- SIP デジット検出および生成パッケージ：SIP メッセージはエンドポイントから SIP プロキシに送信され、DTMF イベントがあったことを、イベントのタイプおよび時間に関する情報とともに示します。
- DTMF (telephone-event コーデック) の RTP ペイロード：RTP パケットには、DTMF が生成されていることを示す情報がヘッダーに含まれています。エンドポイントはこれらのメッセージを理解し、DTMF をローカルに再生します。
- RTP のインバンド ウェーブフォーム：DTMF は音声ウェーブフォームの一部として送信されます。

DTMF インターワーキングの詳細については、次の場所にある『Cisco Unified Border Element (SP Edition) Configuration Guide: Unified Model』の「Implementing Interworking DTMF」の章を参照してください。

http://www.cisco.com/en/US/docs/routers/asr1000/configuration/guide/sbcu/sbc_dtmf.html

RTP telephone-event コーデックと SIP 間のインターワーキング

RTP パケットが telephone-event コーデックを使用した DTMF としてマークされている場合、RTP パケットはストリームから削除されます。DBE は、DTMF イベントが発生したこと、および RTP パケットを SIP DTMF イベントに変換する必要があることを示す H.248 メッセージを、Signaling Border Element (SBE) に送信します。

コールは次の条件を満たしている必要があります。

- telephone-event コーデック (RFC 2833 の場合) が SDP のサイド A にあり、サイド B がないこと。
- dd/etd イベントが、サイド B ではなくサイド A 用にサブスクライブされていること。

SIP と RTP 間の telephone-event コーデックのインターワーキング

エンドポイントが SIP 信号を生成すると、SIP DTMF 信号は完全に帯域外で到達します。SIP DTMF をサポートするエンドポイントは SBE に送信される信号を生成します。次に、SBE はこれが DTMF メッセージであることを認識し、DTMF トーンを RTP ストリームに挿入する必要があることを示す H.248 メッセージを DBE に送信します。DBE は telephone-event コーデックを使用してオーディオ ストリームに RTP DTMF パケットを挿入します。

コールは次の条件を満たしている必要があります。

- telephone-event コーデック (RFC 2833 の場合) が SDP のサイド B にあり、サイド A がないこと。
- dd/etd イベントが、サイド A ではなくサイド B 用にサブスクライブされていること。

RTP telephone-event コーデックから RTP インバンド ウェーブフォーム

RTP パケットが telephone-event コーデックを使用して DTMF としてマークされた後、RTP パケットがストリームから削除され、DTMF ウェーブフォームを含む RTP ストリームは、他のエンドポイントに送信されます。

コールは次の条件を満たしている必要があります。

- telephone-event コーデック (RFC 2833 の場合) が SDP のサイド A にあり、サイド B がないこと。
- dd/etd イベントが、サイド A およびサイド B 用にサブスクライブされていること。

RTP インバンド ウェーブフォームから RTP telephone-event コーデック

DTMF が音声ウェーブフォームの一部として送信された後、RTP パケットがストリームから削除され、DBE はオーディオストリームにペイロードタイプ telephone event で新しい RTP パケットを挿入します。

コールは次の条件を満たしている必要があります。

- telephone-event コーデック (RFC 2833 の場合) が SDP のサイド B にあり、サイド A がないこと。
- dd/etd イベントが、サイド A およびサイド B 用にサブスクライブされていること。

SIP と RTP の間のインバンド ウェーブフォーム

エンドポイントが SIP 信号を生成した後、SIP DTMF 信号は完全に帯域外で到達します。SIP DTMF をサポートするエンドポイントは SBE に送信される信号を生成します。次に、SBE はこれが DTMF メッセージであることを認識し、DTMF トーンを RTP ストリームに挿入する必要があることを示す H.248 メッセージを DBE に送信します。DBE は、DTMF ウェーブフォームを含むストリームを挿入します。

コールは次の条件を満たしている必要があります。

- telephone-event コーデック (RFC 2833 用) がサイド A にもサイド B にもないこと。
- dd/etd イベントが、サイド B 用にサブスクライブされていること。

RTP インバンド ウェーブ フォームから SIP

DTMF が音声ウェーブフォームの一部として送信されると、RTP パケットがストリームから削除され、DBE は、DTMF イベントが発生したこと、および RTP パケットを SIP DTMF イベントに変換する必要があることを示す H.248 メッセージを SBE に送信します。

コールは次の条件を満たしている必要があります。

- telephone-event コーデック (RFC 2833 用) がサイド A にもサイド B にもないこと。
- dd/etd イベントが、サイド A 用にサブスクライブされていること。

コール リカバリ

Cisco IOS XE Release 3.3S から、部分的にクラッシュした SPA DSP のコールは、2.5 秒のコール停止時間内に回復できます。

SPA DSP の一部がクラッシュすると、クラッシュ リカバリ プロセスが実行され、RP は、以前あったすべてのコールを使用して、SPA DSP のクラッシュした部分を再プログラムします。たとえば、単純なトランスコーディングのシナリオで、a-law から u-law へのトランスコーディングは、再プログラムを必要とする最大 129 コールを表すことがあります。

クラッシュする SPA DSP の部分によっては、多くのコンポーネントを回復し、多くのコールを再プログラムする必要があるため、合計リカバリ時間が長くなる場合があります。ただし、すべての回復されたコールのメディア パス全体の中断時間は 2.5 秒未満です。

SPA DSP コール リカバリのいずれのケースでも、コール リカバリは、コールがクラッシュ前に存在していたのと同じ SPA DSP で行われます。コールは別の SPA DSP に移動されません。

SPA DSP 障害コール リカバリは、SPA DSP クラッシュ ダンプがイネーブルになっている場合、ディセーブルまたは無効にできます。これによりコール停止時間が 2.5 秒を超えることがあります。

show voice dsp group all コマンドは、SPA DSP がコール リカバリを実行中かどうかを示します。

```
Router# show voice dsp group all

Show DSP group all

DSP groups on slot 0 bay 0:
dsp 1:
  State: UP
  HA State : DSP_HA_STATE_PENDING1
  Max signal/voice channel: 43/43
  Max credits: 645
  num_of_sig_chnls_allocated: 43
  Transcoding channels allocated: 43
  Group: FLEX_GROUP_XCODE, complexity: LOW
  Shared credits: 0, reserved credits: 645
  Transcoding channels allocated: 24
  Credits used (rounded-up): 360
```



(注) **show voice dsp group all** コマンドでは、リカバリ処理中 (最大数ミリ秒) にのみ「**HA State : DSP_HA_STATE_PENDING1**」という出力が表示されます。

SBC の SPA DSP サービスの設定

ここでは、SBC の SPA DSP サービスの設定に関連する作業について説明します。

- 「DSP ファーム サービスを使用した SPA DSP の設定」 (P.889)
- 「DSP ファーム プロファイルの設定」 (P.889)

DSP ファーム サービスを使用した SPA DSP の設定

DSP サービスの DSP ファーム モードで DSP SPA を設定するには、次の手順を実行します。

手順の概要

1. `configure terminal`
2. `voice-card slot number/subslot number`
3. `dsp services dspfarm`
4. `end`

手順の詳細

	コマンドまたはアクション	目的
ステップ1	<code>configure terminal</code> 例： Router# <code>configure terminal</code>	グローバル コンフィギュレーション モードを開始します。
ステップ2	<code>voice-card slot number/subslot number</code> 例： Router(config)# <code>voice-card 0/2</code>	音声カードのスロット番号を指定し、音声カード インターフェイス コンフィギュレーション モードを開始します。
ステップ3	<code>dsp services dspfarm</code> 例： Router(config-voicecard)# <code>dsp services dspfarm</code>	SPA DSP 音声カードの DSP ファーム サービスを許可します。
ステップ4	<code>end</code> 例： Router(config-voicecard)# <code>end</code>	音声カード インターフェイス コンフィギュレーション モードを終了します。

SPA DSP の DSP ファーム サービスの設定の詳細については、次の場所にある『Cisco ASR 1000 Series Aggregation Services Routers SIP and SPA Software Configuration Guide』の「Configuring the Cisco DSP SPA for ASR 1000 Series」の章を参照してください。

http://www.cisco.com/en/US/docs/interfaces_modules/shared_port_adapters/configuration/ASR1000/asrcfgdsp.html

DSP ファーム プロファイルの設定

DSP ファーム プロファイルを設定するには、次の手順を実行します。

手順の概要

1. **configure terminal**
2. **dspfarm profile** *profile-identifier* {**conference** | **mtp** | **transcode**}
3. **description** *profile-description-text*
4. **codec** *codec-name*
5. **associate application** {**cube** | **sbc** | **sccp**}
6. **maximum session** *number*
7. **no shutdown**
8. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ1	configure terminal 例： Router# configure terminal	グローバル コンフィギュレーション モードを開始します。
ステップ2	dspfarm profile <i>profile-identifier</i> { conference mtp transcode }	指定した DSP ファーム プロファイルの DSP ファーム サービスをイネーブルにし、DSP ファーム プロファイル コンフィギュレーション モードを開始します。 サービス オプションには次のいずれかを指定します。 <ul style="list-style-type: none"> • conference : 会議をイネーブルにします。 • mtp : メディア ターミネーション ポイントをイネーブルにします。 • transcode : 情報のトランスコーディングをイネーブルにします。 (注) Cisco IOS Release 3.2S では、トランスコード サービスだけがサポートされます。
ステップ3	description <i>profile-description-text</i> 例： Router(config-dspfarm-profile)# description <i>enables transcoding</i>	定義したプロファイルの説明を指定します。

コマンドまたはアクション	目的
<p>ステップ4 <code>no codec codec-name</code></p> <p>例： Router(config-dspfarm-profile)# codec g711ulaw Router(config-dspfarm-profile)# codec g711alaw Router(config-dspfarm-profile)# codec g729ar8 Router(config-dspfarm-profile)# codec g729abr8 Router(config-dspfarm-profile)# codec g729r8 Router(config-dspfarm-profile)# codec g723r63 Router(config-dspfarm-profile)# codec ilbc Router(config-dspfarm-profile)# codec gsmamr-nb Router(config-dspfarm-profile)# codec g726r32 Router(config-dspfarm-profile)# codec g729br8</p>	<p>コーデック リストにコーデックを追加したりコーデックを削除します。コーデックは、SBE が認識用にハードコードされているコーデック リスト中に存在する必要があります。</p>
<p>ステップ5 <code>associate application {cube sbc sccp} profile-description-text</code></p> <p>例： Router(config-dspfarm-profile)# associate application sbc</p>	<p>プロファイルにアプリケーションを関連付けます。関連付けることができるアプリケーションは、次のとおりです。</p> <ul style="list-style-type: none"> • cube : Cisco Unified Border Element アプリケーションを DSP ファームの定義済みプロファイルに関連付けます。 • sbc : SBC アプリケーションを DSP ファームの定義済みプロファイルに関連付けます。 • sccp : クライアント プロトコル アプリケーションを DSP ファームの定義済みプロファイルに関連付けます。 <p>(注) sbc アプリケーション キーワードは DSP ファーム プロファイル トランスコード サービスが使用されている場合にだけ使用できます。</p>
<p>ステップ6 <code>maximum session number</code></p> <p>例： Router(config-dspfarm-profile)# maximum session 300</p>	<p>定義されたプロファイルに割り当てることができるセッションの最大数を設定します。セッションの最大数は、ルータ内の SPA DSP 数、および設定されているコーデックによって異なります。23 個の DSP SPA および G711 コーデックのみを備えたフル装備の Cisco ASR 1013 シリーズ ルータでは、セッションの最大数は 20769 です。</p>
<p>ステップ7 <code>no shutdown</code></p> <p>例： Router(config-dspfarm-profile)# no shutdown</p>	<p>DSP ファーム プロファイルをイネーブルまたはディセーブルにします。</p>
<p>ステップ8 <code>end</code></p> <p>例： Router(config-dspfarm-profile)# end</p>	<p>DSP ファーム プロファイルを終了します。</p>

DSP ファーム プロファイルの設定およびステータスの表示例

DSP ファーム プロファイルの作成後に、**show** コマンドを使用して、DSP ファーム プロファイルの設定およびステータスを表示します。次の例は、**show** コマンドの出力を示しています。

```
Router# show running-config
!
voice-card 2/0
```

```

no dspfarm
dsp services dspfarm
!
dspfarm profile 20 transcode
codec g711ulaw
codec g711alaw
codec g729r8
codec g729ar8
codec g729br8
codec g729abr8
rsvp
maximum sessions 5
associate application SBC
!

Router# show dspfarm profile 20

Dspfarm Profile Configuration
Profile ID = 20, Service = TRANSCODING, Resource ID = 1
Profile Description :
Profile Admin State : UP
Profile Operation State : ACTIVE
Application : SBC Status : ASSOCIATED
Resource Provider : FLEX_DSPRM Status : UP
Number of Resource Configured : 5
Number of Resource Available : 5
Codec Configuration
Codec : g729abr8, Maximum Packetization Period : 60
Codec : g711alaw, Maximum Packetization Period : 30
Codec : g711ulaw, Maximum Packetization Period : 30
Codec : g729r8, Maximum Packetization Period : 60
Codec : g729ar8, Maximum Packetization Period : 60
Codec : g729br8, Maximum Packetization Period : 60
RSVP : ENABLED
!

Router# show dspfarm all

DSPFARM Configuration Information:
Admin State: UP, Oper Status: ACTIVE - Cause code: NONE
Transcoding Sessions: 0(Avail: 0), Conferencing Sessions: 2 (Avail: 2)
Trans sessions for mixed-mode conf: 0 (Avail: 0), RTP Timeout: 600
Connection check interval 600 Codec G729 VAD: ENABLED
Total number of active session(s) 0, and connection(s) 0
SLOT DSP CHNL STATUS USE   TYPE SESS-ID CONN-ID PKTS-RXED PKTS-TXED
0  0  1  UP  FREE conf - - - -
0  0  2  UP  FREE conf- - - -
0  0  3  UP  FREE conf - - - -
0  0  4  UP  FREE conf - - - -
0  0  5  UP  FREE conf - - - -
0  0  6  UP  FREE conf - - - -

```

DBE 上の DSP ファーム プロファイルの設定

ここでは、DBE で DSP ファーム プロファイルを設定する方法について説明します。

手順の概要

1. `show dspfarm {all | dsp | profile}`

2. **configure terminal**
3. **sbc *sbc-name* dbe**
4. **associate dspfarm profile {*profile-number* | all}**
5. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ1	<pre>show dspfarm {all dsp profile profile-identifier}</pre> <p>例： Router# show dspfarm profile all</p>	(任意) DSP ファーム設定情報を表示します。 <ul style="list-style-type: none"> • all : DSP ファームのグローバル情報を表示します。 • dsp : すべての DSP に関する情報を表示します。 • profile : DSP ファーム プロファイルを表示します。
ステップ2	<pre>configure terminal</pre> <p>例： Router# configure terminal</p>	グローバル コンフィギュレーション モードを開始します。
ステップ3	<pre>sbc <i>sbc-name</i> dbe</pre> <p>例： Router(config)# sbc mySBC dbe</p>	SBC 上に DBE サービスを作成し、SBC-DBE コンフィギュレーション モードを開始します。
ステップ4	<pre>associate dspfarm profile {<i>profile-number</i> all}</pre> <p>例： Router(config-sbc-dbe)# associate dspfarm profile 1</p>	DSP ファーム プロファイルと SBC を関連付けます。 <ul style="list-style-type: none"> • <i>profile-number</i> : 関連付けるプロファイル番号を指定します。 • all : SBC に関連付けられているプロファイルから、トランスコーディングセッションに最適な DSP ファーム プロファイルを選択することを SBC に許可します。
ステップ5	<pre>end</pre> <p>例： Router(config-sbc-dbe)# end</p>	SBC-DBE コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

SDP の分析を非アクティブにするように SBC を設定

ここでは、トランスコーディングされたコールを設定する SDP の分析を非アクティブにするように SBC を設定する方法について説明します。

手順の概要

1. **show dspfarm {all | dsp | profile}**
2. **configure terminal**
3. **sbc *sbc-name* dbe**
4. **no activate**

SDP の分析を非アクティブにするように SBC を設定

5. `vdbe`
6. `no attach-controllers`
7. `transcoding check {match | none | overlap}`
8. `controller h248 controller-index`
9. `no transrating check`
10. `exit`
11. `attach-controllers`
12. `exit`
13. `activate`
14. `end`
15. `show sbc sbc-name dbc media-stats`

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	<pre>show dspfarm {all dsp profile profile-identifier}</pre> <p>例： Router# show dspfarm profile all</p>	(任意) DSP ファーム設定情報を表示します。 <ul style="list-style-type: none"> • all : DSP ファームのグローバル情報を表示します。 • dsp : すべての DSP に関する情報を表示します。 • profile : DSP ファーム プロファイルを表示します。
ステップ 2	<pre>configure terminal</pre> <p>例： Router# configure terminal</p>	グローバル コンフィギュレーション モードを開始します。
ステップ 3	<pre>sbc sbc-name dbe</pre> <p>例： Router(config)# sbc mySBC dbe</p>	SBC 上で DBE サービスを作成し、SBC-DBE コンフィギュレーション モードを開始します。
ステップ 4	<pre>no activate</pre> <p>例： Router(config-sbc-dbe)# no activate</p>	DBE を非アクティブ化します。
ステップ 5	<pre>vdbe</pre> <p>例： Router(config-sbc-dbe)# vdbe</p>	SBC の仮想データ ボーダー エlement (vDBE) 機能モードを開始します。
ステップ 6	<pre>no attach-controllers</pre> <p>例： Router(config-sbc-dbe-vdbe)# no attach-controllers</p>	no attach-controllers コマンドでコントローラを取り外します。

	コマンドまたはアクション	目的
ステップ7	transcoding check { match none overlap } 例: Router(config-sbc-dbe-vdbe)# transcoding check none	トランスコーディングのオプションを設定します。 <ul style="list-style-type: none"> • check : トランスコーディング チェックをイネーブルにします。 • match : 正確なコーデックの一致チェックを指定します。 • none : コーデックの一致チェックなしを指定します。 • overlap : 重複するコーデックの一致チェックを指定します。
ステップ8	controller h248 { controller-index } 例: Router(config-sbc-dbe-vdbe)# controller h248 1	DBE の H.248 コントローラを設定し、H.248 コントローラ コンフィギュレーション モードを開始します。 ここに示す例では、設定した番号 1 は DBE の H.248 コントローラを識別します。
ステップ9	no transrating check 例: Router(config-sbc-dbe-vdbe-h248)# no transrating check	トランスレーティング オプションをディセーブルにします。 <ul style="list-style-type: none"> • check : トランスレーティング チェックをイネーブルにします。
ステップ10	exit 例: Router(config-sbc-dbe-vdbe-h248)# exit	H.248 コントローラ コンフィギュレーション モードを終了します。
ステップ11	attach-controllers 例: Router(config-sbc-dbe-vdbe)# attach-controllers	DBE を H.248 コントローラに接続します。
ステップ12	exit 例: Router(config-sbc-dbe-vdbe)# exit	vDBE コンフィギュレーション モードを終了します。
ステップ13	activate 例: Router(config-sbc-dbe)# activate	SBC の DBE サービスをアクティブにします。
ステップ14	end 例: Router(config-sbc-dbe)# end	SBC-DBE コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。
ステップ15	show sbc sbc-name dbe media-stats 例: Router# show sbc MySBC dbe media-stats	一般的な DBE 統計情報をリストします。

次に、トランスコーディングされたコール数をリストする **show sbc dbe media-stats** コマンドの出力例を示します。

```
Router# show sbc MySBC dbe media-stats

SBC Service "MySBC"
  Available Bandwidth      = Unlimited
  Available Flows          = 131072
  Available Packet Rate   = Unlimited
  Active Media Flows       = 0
  Peak Media Flows        = 0
  Total Media Flows       = 0
  Active Transcoded Flows = 1
  Peak Transcoded Flows   = 1
  Total Transcoded Flows  = 1
  Active Signaling Flows  = 0
  Peak Signaling Flows    = 0
  Total Signaling Flows   = 0
  SBC Packets Received    = 0
  SBC Octets Received     = 0
  SBC Packets Sent        = 0
  SBC Octets Sent         = 0
  SBC Packets Discarded   = 0
  SBC Octets Discarded    = 0
  No Media Count          = 0
```

Unified SBC の設定

ここでは、SPA DSP 音声カードの SBC を設定するさまざまな方法について説明します。

- [「DSP ファーム プロファイルと Unified SBC の関連付け」 \(P.896\)](#)
- [「トランスコーディングをイネーブルにするための Unified SBC の設定」 \(P.897\)](#)
- [「トランスレーティングをイネーブルにするための Unified SBC の設定」 \(P.902\)](#)
- [「SRTP とトランスコーディングをイネーブルにするための Unified SBC の設定」 \(P.908\)](#)
- [「インバンド DTMF インターワーキングのための Unified SBC の設定」 \(P.914\)](#)

DSP ファーム プロファイルと Unified SBC の関連付け

DSP ファーム プロファイルへの SBC の関連付けは、対応する DSP ファーム プロファイルが作成されてからのみ可能です。グローバル コンフィギュレーション モードで **associate dspfarm profile** コマンドを使用します。

手順の概要

1. **show dspfarm** {all | dsp | profile}
2. **configure terminal**
3. **sbc sbc-name**
4. **associate dspfarm profile** {profile-number | all}
5. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ1	<pre>show dspfarm {all dsp profile profile-identifier}</pre> <p>例： Router# show dspfarm profile all</p>	DSP ファーム設定情報を表示します。 <ul style="list-style-type: none"> • all : DSP ファームのグローバル情報を表示します。 • dsp : すべての DSP に関する情報を表示します。 • profile : DSP ファーム プロファイルを表示します。
ステップ2	<pre>configure terminal</pre> <p>例： Router# configure terminal</p>	グローバル コンフィギュレーション モードをイネーブルにします。
ステップ3	<pre>sbc sbc-name</pre> <p>例： Router(config)# sbc mySBC</p>	SBC で SBC サービスを作成して、SBC コンフィギュレーション モードを開始します。
ステップ4	<pre>associate dspfarm profile {profile-number all}</pre> <p>例： Router(config-sbc)# associate dspfarm profile 1</p>	DSP ファーム プロファイルに SBC を関連付けます。 <ul style="list-style-type: none"> • <i>profile-number</i> : 関連付けるプロファイル番号を指定します。 • all : SBC に関連付けられているプロファイルから、トランスコーディングセッションに最適な DSP ファーム プロファイルを選択することを SBC に許可します。
ステップ5	<pre>end</pre> <p>例： Router(config-sbc-sbe)# end</p>	コンフィギュレーション モードを終了します。

トランスコーディングをイネーブルにするための Unified SBC の設定

ここでは、トランスコーディング機能をイネーブルにするために SBC を設定します。



(注)

この手順では、**caller** コマンドと **callee** コマンドが使用されています。シナリオによっては、**caller** と **callee** のコマンド ペアの代わりに **branch** コマンドを使用できます。**branch** コマンドはリリース 3.5.0 で導入されました。このコマンドの詳細については、「[ダイレクト非制限 CAC ポリシーの設定 \(P.139\)](#)」を参照してください。

手順の概要

1. **configure terminal**
2. **sbc sbc-name**
3. **sbe**
4. **cac-policy-set policy-set-id**
5. **first-cac-scope scope-name**
6. **first-cac-table table-name**

7. `cac-table table-name`
8. `table-type limit list of limit tables`
9. `entry entry-id`
10. `match-value key`
11. `callee-codec-list list-name`
12. `caller-codec-list list-name`
13. `media police strip | reject | degrade`
14. `action cac-complete`
15. `complete`
16. `cac-policy-set global cac-policy-num`
17. `codec-list list-name`
18. `codec codec-nam`
19. `exit`
20. `codec-list list-name`
21. `codec codec-nam`
22. `exit`
23. `end`
24. `show sbc sbc-name sbe call-stats global current5min`

手順の詳細

	コマンドまたはアクション	目的
ステップ1	<code>configure terminal</code> 例: Router# <code>configure terminal</code>	グローバル コンフィギュレーション モードをイネーブ ルにします。
ステップ2	<code>sbc sbc-name</code> 例: Router(config)# <code>sbc mySBC</code>	SBC で SBC サービスを作成して、SBC コンフィギュ レーション モードを開始します。
ステップ3	<code>sbe</code> 例: Router(config-sbc)# <code>sbe</code>	SBC の Signaling Border Element (SBE) 機能モードを開 始します。
ステップ4	<code>cac-policy-set policy-set-id</code> 例: Router(config-sbc-sbe)# <code>cac-policy-set 1</code>	SBE エンティティ内で CAC ポリシーセット コンフィギュ レーション モードを開始して、必要に応じて新規ポリシー セットを作成します。 • <code>policy-set-id</code> : ポリシーセットを特定するためにユー ザによって選択される整数。範囲は 1 ~ 2147483647 です。

コマンドまたはアクション	目的
<p>ステップ5 <code>first-cac-scope scope-name</code></p> <p>例: <pre>Router(config-sbc-sbe-cacpolicy)# first-cac-scope dst-adjacency</pre></p>	<p>ポリシーのアドミッション コントロール ステージで作業を実行するために、制限が最初に定義されるべき範囲を設定します。各 CAC ポリシーには、適用できる範囲があります。この CAC ポリシーは、コール単位で適用できます。</p> <p><code>scope-name</code> には、次のいずれかの値が指定されます。</p> <ul style="list-style-type: none"> • adj-group : 同じ隣接グループのメンバーからのイベントの制限。 • call : 制限が単一コール単位です。 • category : カテゴリ単位の制限 • dst-account : 同じアカウントに送信されるイベントの制限。 • dst-adj-group : 同じ隣接グループに送信されるイベントの制限。 • dst-adjacency : 同じ隣接に送信されるイベントの制限。 • dst-number : 同一隣接グループ番号を持つイベントの制限。 • global : 制限はグローバルで、他のオプションと組み合わせることができません。 • src-account : 同じアカウントからのイベントの制限。 • src-adj-group : 同じ隣接グループからのイベントの制限。 • src-adjacency : 同じ隣接からのイベントの制限。 • src-number : 同じ送信元番号を持つイベントの制限。
<p>ステップ6 <code>first-cac-table table-name</code></p> <p>例: <pre>Router(config-sbc-sbe-cacpolicy)# first-cac-table codec-dst-acc</pre></p>	<p>処理する最初のポリシー テーブルの名前を設定します。CAC ポリシーには、設定済みのテーブルが数多くあります。CAC ポリシーの適用を開始するには、使用される最初のテーブルを定義する必要があります。</p> <ul style="list-style-type: none"> • table-name : 最初に処理すべきアドミッション コントロール テーブル。
<p>ステップ7 <code>cac-table table-name</code></p> <p>例: <pre>Router(config-sbc-sbe-cacpolicy)# cac-table codec-dst-acc</pre></p>	<p>SBE ポリシー セットのコンテキスト内で、アドミッション コントロール テーブルを設定するための CAC テーブルモードを開始します。必要な場合はテーブルを作成します。</p> <ul style="list-style-type: none"> • table-name : アドミッション コントロール テーブル名。

コマンドまたはアクション	目的
<p>ステップ 8 <code>table-type limit list of limit tables</code></p> <p>例 : Router(config-sbc-sbe-cacpolicy-cactable)# table-type limit dst-adjacency</p>	<p>エントリを一致させるのに使用される基準を入力する必要がある新規 CAC 制限テーブルを設定します。</p> <p><code>list of limit tables</code> には、次のいずれかの値を指定できます。</p> <ul style="list-style-type: none"> • account : アカウント名を比較します。 • adj-group : 隣接グループ名を比較します。 • adjacency : 隣接名を比較します。 • all : 比較タイプはありません。すべてのイベントがこのタイプと一致します。 • call-priority : コール プライオリティと比較します。 • category : 番号分析が割り当てられたカテゴリを比較します。 • dst-account : 宛先アカウント名を比較します。 • dst-adj-group : 宛先隣接グループ名を比較します。 • dst-adjacency : 宛先隣接名を比較します。 • dst-prefix : 着信ディジット スtring の先頭を比較します。 • event-type : CAC ポリシー イベント タイプと比較します。 • src-account : 送信元アカウント名を比較します。 • src-adj-group : 送信元隣接グループ名を比較します。 • src-adjacency : 送信元隣接名を比較します。 • src-prefix : 発番号 String の先頭を比較します。
<p>ステップ 9 <code>entry entry-id</code></p> <p>例 : Router(config-sbc-sbe-cacpolicy-cactable)# entry 1</p>	<p>アドミッション コントロール テーブルのエントリを変更するには、CAC テーブル エントリ モードを開始します。</p> <ul style="list-style-type: none"> • entry-id : テーブル エントリを指定します。
<p>ステップ 10 <code>match-value key</code></p> <p>例 : Router(config-sbc-sbe-cacpolicy-cactable-entry) # match-value nava</p>	<p>CAC 制限テーブル タイプにあるエントリの照合値を設定します。</p>
<p>ステップ 11 <code>callee-codec-list list-name</code></p> <p>例 : Router(config-sbc-sbe-cacpolicy-cactable-entry) # callee-codec-list PCMU</p>	<p>コールの着信側レグで使用が許可されているコーデックをリストします。</p>
<p>ステップ 12 <code>caller-codec-list list-name</code></p> <p>例 : Router(config-sbc-sbe-cacpolicy-cactable-entry) # caller-codec-list PCMA</p>	<p>コールのコール元レグで使用が許可されているコーデックをリストします。</p>

	コマンドまたはアクション	目的
ステップ 13	media police strip reject degrade 例 : Router(config-sbc-sbe-cacpolicy-cactable-entry) # media police strip	SBC が、メディア コールの帯域幅限界を超えるメディア ストリームを処理する方法を設定します。
ステップ 14	action cac-complete 例 : Router(config-sbc-sbe-cacpolicy-cactable-entry) # action cac-complete	イベントが一致すると、CAC ポリシーは完了したと見なされます。
ステップ 15	complete 例 : Router(config-sbc-sbe-cacpolicy)# complete	セット全体をコミットしたら、CAC ポリシー セットを完了します。
ステップ 16	cac-policy-set global policy-num 例 : Router(config-sbc-sbe)# cac-policy-set global 1	グローバル CAC ポリシー セットをアクティブにします。CAC ポリシー セットは、 complete 状態でなければデフォルト ポリシーとして割り当てることはできません。 <ul style="list-style-type: none"> • <i>policy-num</i> : 1 ~ 2147483647 のコール ポリシー セット番号。ポリシー セットは、complete 状態でなければデフォルト ポリシーとして割り当てることはできません。
ステップ 17	codec-list list-name 例 : Router(config-sbc-sbe)# codec-list PCMU	コーデック リストを作成し、コーデック リスト コンフィギュレーション モードを開始します。
ステップ 18	codec codec-name 例 : Router(config-sbc-sbe-codec-list)# codec PCMU	コーデックをコーデック リストに追加します。
ステップ 19	exit 例 : Router(config-sbc-sbe-codec-list)# exit	コーデック リスト コンフィギュレーション モードを終了します。
ステップ 20	codec-list list-name 例 : Router(config-sbc-sbe)# codec-list PCMA	コーデック リストを作成し、コーデック リスト コンフィギュレーション モードを開始します。
ステップ 21	codec codec-name 例 : Router(config-sbc-sbe-codec-list)# codec PCMA	コーデックをコーデック リストに追加します。
ステップ 22	exit 例 : Router(config-sbc-sbe-codec-list)# exit	コーデック リスト コンフィギュレーション モードを終了します。

	コマンドまたはアクション	目的
ステップ 23	<code>end</code> 例 : Router(config-sbc-sbe)# end	コンフィギュレーション セッションを終了します。
ステップ 24	<code>show sbc sbc-name sbe call-stats global current5min</code> 例 : Router# show sbc mySBC sbe call-stats global current5min	指定した SBE のすべてのコールの統計情報をリストします。

次に、アクティブなトランスコーディングおよびトランスレーティングされたコール数をリストする **show sbc sbe call-stats global current5min** コマンドの出力例を示します。

```
Router# show sbc mySBC sbe call-stats global current5min
```

```
SBC Service "mySBC"
Statistics for the current 5 mins for global counters
Call count totals:
  Total call attempts = 0
  Total active calls = 1
  Total active IPv6 calls = 0
  Total activating calls = 0
  Total de-activating calls = 0
  Total active emergency calls = 0
  Total active e2 emergency calls = 0
  Total IMS rx active calls = 0
  Total IMS rx call renegotiation attempts = 0
  Total SRTP-RTP interworked calls = 0
  Total active calls not using SRTP = 1
  Total active transcoded calls = 1
  Total active transrated calls = 0
General call failure counters:
  Total call setup failures = 0
  Total active call failures = 0
  Total failed call attempts = 0
  Total failed calls due to update failure = 0
  Total failed calls due to resource failure = 0
  Total failed calls due to congestion = 0
  Total failed calls due to media failure = 0
  Total failed calls due to signaling failure = 0
  Total failed calls due to IMS rx setup failure = 0
  Total failed calls due to IMS rx renegotiation failure = 0
  Total failed calls due to RTP disallowed on call leg = 0
  Total failed calls due to SRTP disallowed on call leg = 0
```

トランスレーティングをイネーブルにするための Unified SBC の設定



(注) トランスレーティングは、コーデック間ではなく、同じコーデックを使用した異なるレートに対してのみサポートされます。したがって、トランスレーティングとトランスコーディングは同時に実行できません。

ここでは、次の方法のいずれかを使用してトランスレーティングをイネーブルにする方法について説明します。

- 「[同じコーデック ポリシーを使用したトランスレーティング](#)」 (P.903)
- 「[新しいコーデック ポリシーを使用したトランスレーティング](#)」 (P.907)

同じコーデック ポリシーを使用したトランスレーティング

この作業では、同じコーデック ポリシーを使用したトランスレーティングをイネーブルにするために SBC を設定します。



(注) この手順では、**caller** コマンドと **callee** コマンドが使用されています。シナリオによっては、**caller** と **callee** のコマンドペアの代わりに **branch** コマンドを使用できます。**branch** コマンドはリリース 3.5.0 で導入されました。このコマンドの詳細については、「[ダイレクト非制限 CAC ポリシーの設定](#)」 (P.139) を参照してください。

手順の概要

1. **configure terminal**
2. **sbc *sbc-name***
3. **sbe**
4. **cac-policy-set *policy-set-id***
5. **first-cac-table *table-name***
6. **cac-table *table-name***
7. **table-type {*policy-set* | *limit* {*list of limit tables*}}**
8. **entry *entry-id***
9. **cac-scope {*list of scope options*}**
10. **callee ptime *0-100***
11. **caller ptime *0-100***
12. **media police strip | reject | degrade**
13. **action cac complete**
14. **complete**
15. **cac-policy-set global *cac-policy-num***
16. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	<code>configure terminal</code> 例: Router# <code>configure terminal</code>	グローバル コンフィギュレーション モードをイネーブルにします。
ステップ 2	<code>sbc sbc-name</code> 例: Router(config)# <code>sbc mySBC</code>	SBC で SBC サービスを作成して、SBC コンフィギュレーション モードを開始します。
ステップ 3	<code>sbe</code> 例: Router(config-sbc)# <code>sbe</code>	SBC の SBE 機能モードを開始します。
ステップ 4	<code>cac-policy-set policy-set-id</code> 例: Router(config-sbc-sbe)# <code>cac-policy-set 1</code>	SBE エンティティ内で CAC ポリシー セット コンフィギュレーション モードを開始して、必要に応じて新規ポリシー セットを作成します。
ステップ 5	<code>first-cac-table table-name</code> 例: Router(config-sbc-sbe-cacpolicy)# <code>first-cac-table Transrate</code>	処理する最初のポリシー テーブルの名前を設定します。CAC ポリシーには、設定済みのテーブルが数多くあります。CAC ポリシーの適用を開始するには、使用される最初のテーブルを定義する必要があります。 <ul style="list-style-type: none"><code>table-name</code> : 最初に処理すべきアドミッション コントロール テーブル。
ステップ 6	<code>cac-table table-name</code> 例: Router(config-sbc-sbe-cacpolicy)# <code>cac-table Transrate</code>	SBE ポリシー セットのコンテキスト内で、アドミッション コントロール テーブルを設定するための CAC テーブル モードを開始します。必要な場合はテーブルを作成します。 <ul style="list-style-type: none"><code>table-name</code> : アドミッション コントロール テーブル名。

コマンドまたはアクション	目的
<p>ステップ7 <code>table-type {policy-set limit {list of limit tables}}</code></p> <p>例: Router(config-sbc-sbe-cacpolicy-cactable)# table-type policy-set</p>	<p>SBC ポリシー セットのコンテキスト内で CAC テーブルのテーブル タイプを設定します。</p> <p><i>list of limit tables</i> には、次のいずれかの値を指定できます。</p> <ul style="list-style-type: none"> • account : アカウント名を比較します。 • adj-group : 隣接グループ名を比較します。 • adjacency : 隣接名を比較します。 • all : 比較タイプはありません。すべてのイベントがこのタイプと一致します。 • call-priority : コール プライオリティと比較します。 • category : 番号分析が割り当てられたカテゴリを比較します。 • dst-account : 宛先アカウント名を比較します。 • dst-adj-group : 宛先隣接グループ名を比較します。 • dst-adjacency : 宛先隣接名を比較します。 • dst-prefix : 着信ディジット スtringの先頭を比較します。 • event-type : CAC ポリシー イベント タイプと比較します。 • src-account : 送信元アカウント名を比較します。 • src-adj-group : 送信元隣接グループ名を比較します。 • src-adjacency : 送信元隣接名を比較します。 • src-prefix : 発番号Stringの先頭を比較します。 <p>機能は、個々の隣接ごとに行う場合と同じように、CAC 設定から隣接グループごとにイネーブルまたはディセーブルにできます。adj-group テーブル タイプは、送信元または宛先の隣接グループで一致します。</p> <p>policy-set キーワードが指定されている場合、cac-scope コマンドを使用して、CAC ポリシー セット テーブル内で制限が適用される、各エントリ内の範囲を設定します。</p>
<p>ステップ8 <code>entry entry-id</code></p> <p>例: Router(config-sbc-sbe-cacpolicy-cactable)# entry 1</p>	<p>アドミッション コントロール テーブルのエントリを作成または変更するには、CAC テーブル エントリ モードを開始します。</p>

	コマンドまたはアクション	目的
ステップ9	<p><code>cac-scope {list of scope options}</code></p> <p>例 : Router(config-sbc-sbe-cacpolicy-cactable-entry) # cac-scope call</p>	<p>CAC の制限がポリシー セット テーブル内の各エントリに適用される範囲の選択をイネーブルにします。</p> <p><i>list of scope options</i> : イベント照合に使用される次のいずれかのストリングを指定します。</p> <ul style="list-style-type: none"> • <i>account</i> : 同じアカウントからのイベント。 • <i>adjacency</i> : 同じ隣接からのイベント。 • <i>adj-group</i> : 同じ隣接グループのメンバーからのイベント。 • <i>call</i> : スコープの限界は単一コール単位です。 • <i>category</i> : 同じカテゴリを持つイベント。 • <i>dst-account</i> : 同じアカウントに送信されるイベント。 • <i>dst-adj-group</i> : 同じ隣接グループに送信されるイベント。 • <i>dst-adjacency</i> : 同じ隣接に送信されるイベント。 • <i>dst-number</i> : 宛先が同じイベント。 • <i>global</i> : スコープの限界はグローバルです。 • <i>src-account</i> : 同じアカウントからのイベント。 • <i>src-adj-group</i> : 同じ隣接グループからのイベント。 • <i>src-adjacency</i> : 同じ隣接からのイベント。 • <i>src-number</i> : 同じ送信元番号を持つイベント。
ステップ10	<p><code>callee ptime <0-100></code></p> <p>例 : Router(config-sbc-sbe-cacpolicy-cactable-entry) # callee ptime 30</p>	<p>この CAC エントリを使用してコールすることを強制されている着信側のパケット化時間を設定します。</p> <p>デフォルトでは 0 ミリ秒が設定されており、トランスレーティングが実行されません。</p>
ステップ11	<p><code>caller ptime <0-100></code></p> <p>例 : Router(config-sbc-sbe-cacpolicy-cactable-entry) # caller ptime 40</p>	<p>この CAC エントリを使用してコールすることを強制されている発信側のパケット化時間を設定します。</p> <p>デフォルトでは 0 ミリ秒が設定されており、トランスレーティングが実行されません。</p>
ステップ12	<p><code>media police strip reject degrade</code></p> <p>例 : Router(config-sbc-sbe-cacpolicy-cactable-entry) # media police strip</p>	<p>SBC が、メディア コールの帯域幅限界を超えるメディア ストリームを処理する方法を設定します。</p>
ステップ13	<p><code>action cac-complete</code></p> <p>例 : Router(config-sbc-sbe-cacpolicy-cactable-entry) # action cac complete</p>	<p>イベントが一致する場合は、この CAC ポリシーは終了します。</p>

	コマンドまたはアクション	目的
ステップ 14	<code>complete</code> 例： Router(config-sbc-sbe-cacpolicy)# complete	セット全体をコミットしたら、CAC ポリシー セットを完了します。
ステップ 15	<code>cac-policy-set global policy-num</code> 例： Router(config-sbc-sbe)# cac-policy-set global 1	グローバル CAC ポリシー セットをアクティブにします。CAC ポリシー セットは、 complete 状態でなければデフォルト ポリシーとして割り当てることはできません。 <ul style="list-style-type: none"> <code>policy-num</code> : 1 ~ 2147483647 のコール ポリシー セット番号。ポリシー セットは、complete 状態でなければデフォルト ポリシーとして割り当てることはできません。
ステップ 16	<code>end</code> 例： Router(config-sbc-sbe-cacpolicy-cactable-entry)# end	CAC コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

新しいコーデック ポリシーを使用したトランスレーティング

ここでは、トランスレーティング機能をイネーブルにするために SBC を設定します。これは、トランスレーティングを設定するために「[同じコーデック ポリシーを使用したトランスレーティング \(P.903\)](#)」で説明したメカニズムの代替メカニズムです。

手順の概要

1. `configure terminal`
2. `sbc sbc-name`
3. `sbe`
4. `codec list list-name`
5. `policy {minimum | transrating}`
6. `codec codec-name packetization-period packet-period [priority priority-value]`
7. `end`

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	<code>configure terminal</code> 例： Router# <code>configure terminal</code>	グローバル コンフィギュレーション モードをイネーブルにします。
ステップ 2	<code>sbc sbc-name</code> 例： Router(config)# <code>sbc mySBC</code>	SBC で SBC サービスを作成して、SBC コンフィギュレーション モードを開始します。

Unified SBC の設定

	コマンドまたはアクション	目的
ステップ3	sbe 例： Router(config-sbc)# sbe	SBC の SBE 機能モードを開始します。
ステップ4	codec list list-name 例： Router(config-sbc-sbe)# codec list PCMU	コーデック リストを作成しコーデック リスト モードを開始します。
ステップ5	policy {minimum transrating} 例： Router(config-sbc-sbe-codec-list)# policy minimum	次のいずれかに指定されたパケット化期間ポリシーを設定します。 <ul style="list-style-type: none"> • minimum : パケット化期間は最小値です。 • transrating : パケット化期間は、トランスレーティングです。
ステップ6	codec codec-name packetization-period packet-period [priority priority-value] 例： Router(config-sbc-sbe-codec-list)# codec PCMU packetization-period 200 priority 1	コーデックをコーデック リストに追加し、コーデックの最小パケット化期間とプライオリティ値を設定します。
ステップ7	end 例： Router(config-sbc-sbe-codec-list)# end	CAC コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

SRTP とトランスコーディングをイネーブルにするための Unified SBC の設定

Secure Real-Time Transport Protocol (SRTP) は、トランスコーディングとは無関係ですが、同時に両方を使用するように設定できます。

ここでは、SRTP とトランスコーディング機能をイネーブルにするように Unified SBC を設定します。



(注)

この手順では、**caller** コマンドと **callee** コマンドが使用されています。シナリオによっては、**caller** と **callee** のコマンド ペアの代わりに **branch** コマンドを使用できます。**branch** コマンドはリリース 3.5.0 で導入されました。このコマンドの詳細については、「[ダイレクト非制限 CAC ポリシーの設定 \(P.139\)](#)」を参照してください。

手順の概要

1. **configure terminal**
2. **sbc sbc-name**
3. **sbe**
4. **cac-policy-set policy-set-id**
5. **first-cac-table table-name**

6. **cac-table** *table-name*
7. **table-type** {*policy-set* | **limit** {*list of limit tables*}}
8. **entry** *entry-id*
9. **cac-scope** {*list of scope options*}
10. **srtp support** **allow**
11. **srtp caller** **forbid** | **mandate** | **allow** | **prefer**
12. **srtp callee** **forbid** | **mandate** | **allow** | **prefer**
13. **srtp interworking** **forbid** | **allow**
14. **srtp media interworking** **forbid** | **allow**
15. **action next-table** *goto-table-name*
16. **exit**
17. **exit**
18. **cac-table** *table-name*
19. **table-type** **limit** *list of limit tables*
20. **entry** *entry-id*
21. **match-value** *key*
22. **callee-codec-list** *list-name*
23. **action cac-complete**
24. **complete**
25. **cac-policy-set** **global** *cac-policy-num*
26. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ1	configure terminal 例： Router# configure terminal	グローバル コンフィギュレーション モードをイネーブルにします。
ステップ2	sbc <i>sbc-name</i> 例： Router(config)# sbc mySBC	SBC で SBC サービスを作成して、SBC コンフィギュレーション モードを開始します。
ステップ3	sbe 例： Router(config-sbc)# sbe	SBC の SBE 機能モードを開始します。

	コマンドまたはアクション	目的
ステップ4	<p>cac-policy-set <i>policy-set-id</i></p> <p>例: Router(config-sbc-sbe)# cac-policy-set 3</p>	SBE エンティティ内で CAC ポリシー セット コンフィギュレーション モードを開始して、必要に応じて新規ポリシー セットを作成します。
ステップ5	<p>first-cac-table <i>table-name</i></p> <p>例: Router(config-sbc-sbe-cacpolicy)# first-cac-table C3</p>	<p>処理する最初のポリシー テーブルの名前を設定します。CAC ポリシーには、設定済みのテーブルが数多くあります。CAC ポリシーの適用を開始するには、使用される最初のテーブルを定義する必要があります。</p> <ul style="list-style-type: none"> <i>table-name</i> : 最初に処理すべきアドミッション コントロール テーブル。
ステップ6	<p>cac-table <i>table-name</i></p> <p>例: Router(config-sbc-sbe-cacpolicy)# cac-table C3</p>	<p>SBE ポリシー セットのコンテキスト内で、アドミッション コントロール テーブルを設定するための CAC テーブル モードを開始します。必要な場合はテーブルを作成します。</p> <ul style="list-style-type: none"> <i>table-name</i> : アドミッション コントロール テーブル名。

コマンドまたはアクション	目的
<p>ステップ7 <code>table-type {policy-set limit {list of limit tables}}</code></p> <p>例: Router(config-sbc-sbe-cacpolicy-cactable)# table-type policy-set</p>	<p>SBC ポリシー セットのコンテキスト内で CAC テーブルのテーブル タイプを設定します。</p> <p><i>list of limit tables</i> には、次のいずれかの値を指定できます。</p> <ul style="list-style-type: none"> • account : アカウント名を比較します。 • adj-group : 隣接グループ名を比較します。 • adjacency : 隣接名を比較します。 • all : 比較タイプはありません。すべてのイベントがこのタイプと一致します。 • call-priority : コール プライオリティと比較します。 • category : 番号分析が割り当てられたカテゴリを比較します。 • dst-account : 宛先アカウント名を比較します。 • dst-adj-group : 宛先隣接グループ名を比較します。 • dst-adjacency : 宛先隣接名を比較します。 • dst-prefix : 着信ディジット スtring の先頭を比較します。 • event-type : CAC ポリシー イベント タイプと比較します。 • src-account : 送信元アカウント名を比較します。 • src-adj-group : 送信元隣接グループ名を比較します。 • src-adjacency : 送信元隣接名を比較します。 • src-prefix : 発番号 String の先頭を比較します。 <p>機能は、個々の隣接ごとに行う場合と同じように、CAC 設定から隣接グループごとにイネーブルまたはディセーブルにできます。adj-group テーブル タイプは、送信元または宛先の隣接グループで一致します。</p> <p>policy-set キーワードが指定されている場合、cac-scope コマンドを使用して、CAC ポリシー セット テーブル内で制限が適用される、各エントリ内の範囲を設定します。</p>
<p>ステップ8 <code>entry entry-id</code></p> <p>例: Router(config-sbc-sbe-cacpolicy-cactable)# entry 1</p>	<p>アドミッション コントロール テーブル内のエントリを作成または変更するモードを開始します。</p>

コマンドまたはアクション	目的
<p>ステップ9 <code>cac-scope {list of scope options}</code></p> <p>例 : Router(config-sbc-sbe-cacpolicy-cactable-entry) # cac-scope global</p>	<p>CAC の制限がポリシー セット テーブル内の各エンタリに適用される範囲を選択します。</p> <p><i>list of scope options</i> : イベント照合に使用される次のいずれかのストリングを指定します。</p> <ul style="list-style-type: none"> • <i>account</i> : 同じアカウントからのイベント。 • <i>adjacency</i> : 同じ隣接からのイベント。 • <i>adj-group</i> : 同じ隣接グループのメンバーからのイベント。 • <i>call</i> : スコープの限界は単一コール単位です。 • <i>category</i> : 同じカテゴリを持つイベント。 • <i>dst-account</i> : 同じアカウントに送信されるイベント。 • <i>dst-adj-group</i> : 同じ隣接グループに送信されるイベント。 • <i>dst-adjacency</i> : 同じ隣接に送信されるイベント。 • <i>dst-number</i> : 宛先が同じイベント。 • <i>global</i> : スコープの限界はグローバルです。 • <i>src-account</i> : 同じアカウントからのイベント。 • <i>src-adj-group</i> : 同じ隣接グループからのイベント。 • <i>src-adjacency</i> : 同じ隣接からのイベント。 • <i>src-number</i> : 同じ送信元番号を持つイベント。
<p>ステップ10 <code>srtp support allow</code></p> <p>例 : Router(config-sbc-sbe-cacpolicy-cactable-entry) # srtp support allow</p>	<p>SRTP サポートを設定します。</p>
<p>ステップ11 <code>srtp caller forbid mandate allow prefer</code></p> <p>例 : Router(config-sbc-sbe-cacpolicy-cactable-entry) # srtp caller mandate</p>	<p>次のいずれかの SRTP 設定を使用して、コールの発信側の SRTP を設定します。</p> <ul style="list-style-type: none"> • forbid : SRTP はコールの発信側でサポートされません。 • mandate : SRTP はコールの発信側で必須です。 • allow : SRTP はコールの発信側で任意です。 • prefer : この隣接では SRTP が優先されます。RTP および SRTP の両方が受信で許可されますが、送信では SRTP のみが提供されます。

	コマンドまたはアクション	目的
ステップ 12	<pre>srtp callee forbid mandate allow prefer</pre> <p>例:</p> <pre>Router(config-sbc-sbe-cacpolicy-cactable-entry) # srtp callee mandate</pre>	<p>コールの着信側の SRTP を設定します。</p> <ul style="list-style-type: none"> • forbid : SRTP はコールの着信側でサポートされません。 • mandate : SRTP はコールの着信側で必須です。 • allow : SRTP はコールの着信側で任意です。 • prefer : この隣接では SRTP が優先されます。RTP および SRTP の両方が受信で許可されますが、送信では SRTP のみが提供されます。
ステップ 13	<pre>srtp interworking forbid allow</pre> <p>例:</p> <pre>Router(config-sbc-sbe-cacpolicy-cactable-entry) # srtp interworking allow</pre>	<p>SRTP と RTP 間のインターワーキングを設定します。</p> <ul style="list-style-type: none"> • forbid : コールに対し SRTP と RTP 間のインターワーキングを禁止します。 • allow : コールに対し SRTP と RTP 間のインターワーキングを許可します。
ステップ 14	<pre>srtp media interworking forbid allow</pre> <p>例:</p> <pre>Router(config-sbc-sbe-cacpolicy-cactable-entry) # srtp media interworking allow</pre>	<p>SRTP と RTP 間のメディア インターワーキングを設定します。</p> <ul style="list-style-type: none"> • forbid : コールに対し SRTP と RTP 間のメディア インターワーキングを禁止します。 • allow : コールに対し SRTP と RTP 間のメディア インターワーキングを許可します。
ステップ 15	<pre>action next-table goto-table-name</pre> <p>例:</p> <pre>Router(config-sbc-sbe-cacpolicy-cactable-entry) # action next-table xcode</pre>	<p>このルーティング エントリが選択された場合に実行するアクションを設定します。</p> <ul style="list-style-type: none"> • goto-table-name : イベントがエントリに一致する場合に処理する次のルーティング テーブルを指定します。
ステップ 16	<pre>exit</pre> <p>例:</p> <pre>Router(config-sbc-sbe-cacpolicy-cactable-entry) # exit</pre>	<p>CAC テーブル エントリ コンフィギュレーション モードを終了します。</p>
ステップ 17	<pre>exit</pre> <p>例:</p> <pre>Router(config-sbc-sbe-cacpolicy-cactable)# exit</pre>	<p>CAC テーブル コンフィギュレーション モードを終了します。</p>
ステップ 18	<pre>cac-table table-name</pre> <p>例:</p> <pre>Router(config-sbc-sbe-cacpolicy)# cac-table xcode</pre>	<p>SBE ポリシー セットのコンテキスト内で、アドミッション コントロール テーブルを設定するための CAC テーブル モードを開始します。必要な場合はテーブルを作成します。</p> <ul style="list-style-type: none"> • table-name : アドミッション コントロール テーブル名。
ステップ 19	<pre>table-type limit list of limit tables</pre> <p>例:</p> <pre>Router(config-sbc-sbe-cacpolicy-cactable)# table-type limit dst-adjacency</pre>	<p>match-value コマンドで照合するテーブル タイプの制限を設定します。ここに示す例では、次のテーブル タイプを使用します。</p> <ul style="list-style-type: none"> • dst-adjacency : 宛先隣接名を比較します。

Unified SBC の設定

	コマンドまたはアクション	目的
ステップ 20	<code>entry entry-id</code> 例： Router(config-sbc-sbe-cacpolicy-cactable)# entry 1	アドミッション コントロール テーブルのエントリを変更するには、CAC テーブル エントリ モードを開始します。 • <i>entry-id</i> : テーブル エントリを指定します。
ステップ 21	<code>match-value key</code> 例： Router(config-sbc-sbe-cacpolicy-cactable-entry) # match-value nav4B	コール アドミッション制御 (CAC) 制限テーブルにあるエントリの照合値を設定します。 • <i>key</i> : イベントの照合に使用するキーワードを指定します。キーの形式は、 <i>table-type</i> 制限によって決定されます。
ステップ 22	<code>callee-codec-list list-name</code> 例： Router(config-sbc-sbe-cacpolicy-cactable-entry) # callee-codec-list PCMU	コールの着信側レグで使用が許可されているコーデックをリストします。 • <i>list-name</i> : コーデック リストの名前を指定します。最大サイズは 30 文字です。
ステップ 23	<code>action cac-complete</code> 例： Router(config-sbc-sbe-cacpolicy-cactable-entry) # action cac-complete	イベントが一致する場合は、この CAC ポリシーは終了します。
ステップ 24	<code>complete</code> 例： Router(config-sbc-sbe-cacpolicy)# complete	セット全体をコミットしたら、CAC ポリシー セットを完了します。
ステップ 25	<code>cac-policy-set global policy-num</code> 例： Router(config-sbc-sbe)# cac-policy-set global 3	グローバル CAC ポリシー セットをアクティブにします。CAC ポリシー セットは、 complete 状態でなければデフォルト ポリシーとして割り当てることはできません。 • <i>policy-num</i> : 1 ~ 2147483647 のコール ポリシー セット番号。ポリシー セットは、 complete 状態でなければデフォルト ポリシーとして割り当てることはできません。
ステップ 26	<code>end</code> 例： Router(config-sbc-sbe-cacpolicy-cactable-entry) # end	CAC コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

インバンド DTMF インターワーキングのための Unified SBC の設定

SPA DSP は、インバンドと呼ばれる、リアルタイム転送プロトコル (RTP) ストリームで再生する DTMF トーンを検出するために使用できます。インバンド DTMF インターワーキングは SPA DSP リソースを使用し、プレーン コールおよびトランスコーディングされたコールに使用できます。



(注)

この手順では、**caller** コマンドと **callee** コマンドが使用されています。シナリオによっては、**caller** と **callee** のコマンド ペアの代わりに **branch** コマンドを使用できます。**branch** コマンドはリリース 3.5.0 で導入されました。このコマンドの詳細については、「[ダイレクト非制限 CAC ポリシーの設定](#)」(P.139) を参照してください。

手順の概要

1. **configure terminal**
2. **sbc *sbc-name***
3. **sbe**
4. **cac-policy-set *policy-set-id***
5. **first-cac-table *table-name***
6. **cac-table *table-name***
7. **table-type {*policy-set* | *limit* {*list of limit tables*}}**
8. **entry *entry-id***
9. **cac-scope {*list of scope options*}**
10. **callee inband-dtmf-mode always**
11. **caller inband-dtmf-mode never**
12. **action next-table *goto-table-name***
13. **complete**
14. **cac-policy-set global *cac-policy-num***
15. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ1	configure terminal 例： Router# configure terminal	グローバル コンフィギュレーション モードをイネーブルにします。
ステップ2	sbc <i>sbc-name</i> 例： Router(config)# sbc mySBC	SBC で SBC サービスを作成して、SBC コンフィギュレーション モードを開始します。
ステップ3	sbe 例： Router(config-sbc)# sbe	SBC の SBE 機能モードを開始します。

	コマンドまたはアクション	目的
ステップ4	<p><code>cac-policy-set policy-set-id</code></p> <p>例: Router(config-sbc-sbe)# cac-policy-set 4</p>	SBE エンティティ内で CAC ポリシー セット コンフィギュレーション モードを開始して、必要に応じて新規ポリシー セットを作成します。
ステップ5	<p><code>first-cac-table table-name</code></p> <p>例: Router(config-sbc-sbe-cacpolicy)# first-cac-table C4</p>	<p>処理する最初のポリシー テーブルの名前を設定します。CAC ポリシーには、設定済みのテーブルが数多くあります。CAC ポリシーの適用を開始するには、使用される最初のテーブルを定義する必要があります。</p> <ul style="list-style-type: none"> <i>table-name</i> : 最初に処理すべきアドミッション コントロール テーブル。
ステップ6	<p><code>cac-table table-name</code></p> <p>例: Router(config-sbc-sbe-cacpolicy)# cac-table C4</p>	<p>SBE ポリシー セットのコンテキスト内で、アドミッション コントロール テーブルを設定するための CAC テーブル モードを開始します。必要な場合はテーブルを作成します。</p> <ul style="list-style-type: none"> <i>table-name</i> : アドミッション コントロール テーブル名。

コマンドまたはアクション	目的
<p>ステップ7 <code>table-type {policy-set limit {list of limit tables}}</code></p> <p>例: Router(config-sbc-sbe-cacpolicy-cactable)# table-type policy-set</p>	<p>SBC ポリシー セットのコンテキスト内で CAC テーブルのテーブル タイプを設定します。</p> <p><i>list of limit tables</i> には、次のいずれかの値を指定できます。</p> <ul style="list-style-type: none"> • account : アカウント名を比較します。 • adj-group : 隣接グループ名を比較します。 • adjacency : 隣接名を比較します。 • all : 比較タイプはありません。すべてのイベントがこのタイプと一致します。 • call-priority : コール プライオリティと比較します。 • category : 番号分析が割り当てられたカテゴリを比較します。 • dst-account : 宛先アカウント名を比較します。 • dst-adj-group : 宛先隣接グループ名を比較します。 • dst-adjacency : 宛先隣接名を比較します。 • dst-prefix : 着信ディジット スtring の先頭を比較します。 • event-type : CAC ポリシー イベント タイプと比較します。 • src-account : 送信元アカウント名を比較します。 • src-adj-group : 送信元隣接グループ名を比較します。 • src-adjacency : 送信元隣接名を比較します。 • src-prefix : 発番号 String の先頭を比較します。 <p>機能は、個々の隣接ごとに行う場合と同じように、CAC 設定から隣接グループごとにイネーブルまたはディセーブルにできます。adj-group テーブル タイプは、送信元または宛先の隣接グループで一致します。</p> <p>policy-set キーワードが指定されている場合、cac-scope コマンドを使用して、CAC ポリシー セット テーブル内で制限が適用される、各エントリ内の範囲を設定します。</p>
<p>ステップ8 <code>entry entry-id</code></p> <p>例: Router(config-sbc-sbe-cacpolicy-cactable)# entry 1</p>	<p>アドミッション コントロール テーブルのエントリを作成または変更するには、CAC テーブル エントリ モードを開始します。</p>

コマンドまたはアクション	目的
<p>ステップ9 <code>cac-scope {list of scope options}</code></p> <p>例: Router(config-sbc-sbe-cacpolicy-cactable-entry) # cac-scope global</p>	<p>CAC の制限がポリシー セット テーブル内の各エントリに適用される範囲を選択します。</p> <p><i>list of scope options</i> : イベント照合に使用される次のいずれかのストリングを指定します。</p> <ul style="list-style-type: none"> • <i>account</i> : 同じアカウントからのイベント。 • <i>adjacency</i> : 同じ隣接からのイベント。 • <i>adj-group</i> : 同じ隣接グループのメンバーからのイベント。 • <i>call</i> : スコープの限界は単一コール単位です。 • <i>category</i> : 同じカテゴリを持つイベント。 • <i>dst-account</i> : 同じアカウントに送信されるイベント。 • <i>dst-adj-group</i> : 同じ隣接グループに送信されるイベント。 • <i>dst-adjacency</i> : 同じ隣接に送信されるイベント。 • <i>dst-number</i> : 宛先が同じイベント。 • <i>global</i> : スコープの限界はグローバルです。 • <i>src-account</i> : 同じアカウントからのイベント。 • <i>src-adj-group</i> : 同じ隣接グループからのイベント。 • <i>src-adjacency</i> : 同じ隣接からのイベント。 • <i>src-number</i> : 同じ送信元番号を持つイベント。
<p>ステップ10 <code>callee inband-dtmf-mode {always inherit maybe never}</code></p> <p>例: Router(config-sbc-sbe-cacpolicy-cactable-entry) # callee inband-dtmf-mode always</p>	<p>着信側の DTMF インバンド モードを設定します。</p> <ul style="list-style-type: none"> • always : インバンド DTMF トーンはエンドポイントで常に使用されています。 • inherit : エンドポイントのインバンド DTMF モードはこの CAC エントリの影響を受けません。 • maybe : DTMF の代替形式が使用されていることがシグナリングによって示されない限り、インバンド DTMF トーンはエンドポイントで使用されます。 • never : エンドポイントは、インバンド DTMF を使用しません。
<p>ステップ11 <code>caller inband-dtmf-mode {always inherit maybe never}</code></p> <p>例: Router(config-sbc-sbe-cacpolicy-cactable-entry) # caller inband-dtmf-mode never</p>	<p>発信側の DTMF インバンド モードを設定します。</p> <ul style="list-style-type: none"> • always : インバンド DTMF トーンはエンドポイントで常に使用されています。 • inherit : エンドポイントのインバンド DTMF モードはこの CAC エントリの影響を受けません。 • maybe : DTMF の代替形式が使用されていることがシグナリングによって示されない限り、インバンド DTMF トーンはエンドポイントで使用されます。 • never : エンドポイントは、インバンド DTMF を使用しません。

	コマンドまたはアクション	目的
ステップ 12	<pre>action next-table goto-table-name</pre> <p>例 :</p> <pre>Router(config-sbc-sbe-cacpolicy-cactable-entry) # action next-table xcode</pre>	<p>このルーティング エントリが選択された場合に実行するアクションを設定します。</p> <ul style="list-style-type: none"> <i>goto-table-name</i> : イベントがエントリに一致する場合に処理する次のルーティング テーブルを指定します。
ステップ 13	<pre>complete</pre> <p>例 :</p> <pre>Router(config-sbc-sbe-cacpolicy)# complete</pre>	<p>セット全体をコミットしたら、CAC ポリシー セットを完了します。</p>
ステップ 14	<pre>cac-policy-set global policy-num</pre> <p>例 :</p> <pre>Router(config-sbc-sbe)# cac-policy-set global 4</pre>	<p>グローバル CAC ポリシー セットをアクティブにします。CAC ポリシー セットは、complete 状態でなければデフォルト ポリシーとして割り当ててはできません。</p> <ul style="list-style-type: none"> <i>policy-num</i> : 1 ~ 2147483647 のコール ポリシー セット番号。ポリシー セットは、complete 状態でなければデフォルト ポリシーとして割り当ててはできません。
ステップ 15	<pre>end</pre> <p>例 :</p> <pre>Router(config-sbc-sbe-cacpolicy-cactable-entry) # end</pre>	<p>CAC コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。</p>

SBC の SPA DSP サービスの設定例

ここでは、次の例を示します。

- 「[設定例：SPA DSP の DSP ファーム サービスのイネーブル化](#)」(P.919)
- 「[DSP ファーム プロファイルの設定例](#)」(P.919)

設定例：SPA DSP の DSP ファーム サービスのイネーブル化

次に、SPA DSP で DSP ファーム サービスをイネーブルにする例を示します。

```
enable
configure terminal
voice-card 0/2
dsp services dspfarm
end
```

DSP ファーム プロファイルの設定例

次に、DSP ファーム プロファイルを設定する例を示します。

```
enable
configure terminal
```

```
dspfarm profile 1 transcode
description enables transcoding
  codec g711ulaw
  codec g711alaw
  codec g729ar8
  codec g729abr8
  codec g729r8
  codec g723r63
  codec ilbc
  codec gsmamr-nb
  codec g726r32
  codec g729br8
associate application sbc
maximum session 300
end
```

DBE の DSP ファーム プロファイルの設定例

次に、DBE で DSP ファーム プロファイルを設定する例を示します。

```
enable
configure terminal
sbc MySBC dbe
  associate dspfarm profile 1
end
```

SDP の分析を非アクティブにするように SBC を設定する例

次に、トランスコーディングされたコールを設定する SDP 分析を非アクティブにするように SBC を設定する例を示します。

```
enable
configure terminal
sbc mySBC dbe
  no activate
  vdbe
  no attach-controllers
  transcoding check none
  controller h248 1
  no transrating check
  exit
attach-controllers
exit
activate
end
```

Unified SBC の設定例

ここでは、次の例を示します。

- 「[設定例：DSP ファーム プロファイルと Unified SBC の関連付け](#)」(P.921)
- 「[トランスコーディングをイネーブルにするための Unified SBC の設定例](#)」(P.921)
- 「[トランスレーティングをイネーブルにするための Unified SBC の設定例](#)」(P.921)
- 「[SRTP とトランスコーディングをイネーブルにするための Unified SBC の設定例](#)」(P.922)

- 「インバンド DTMF インターワーキングのための Unified SBC の設定例」 (P.923)

設定例 : DSP ファーム プロファイルと Unified SBC の関連付け

次に、Unified SBC を DSP ファーム プロファイルに関連付ける例を示します。

```
enable
configure terminal
  sbc mySBC
  associate dspfarm profile 1
end
```

トランスコーディングをイネーブルにするための Unified SBC の設定例

次に、トランスコーディングをイネーブルにするように Unified SBC を設定する例を示します。



(注)

この手順では、**caller** コマンドと **callee** コマンドが使用されています。シナリオによっては、**caller** と **callee** のコマンド ペアの代わりに **branch** コマンドを使用できます。**branch** コマンドはリリース 3.5.0 で導入されました。このコマンドの詳細については、「[ダイレクト非制限 CAC ポリシーの設定](#)」(P.139) を参照してください。

```
enable
configure terminal
  sbc mySBC
  sbe
  cac-policy-set 1
  first-cac-scope dst-adjacency
  first-cac-table codec-dst-acc
  cac-table codec-dst-acc
  table-type limit dst-adjacency
  entry 1
  match-value nava
  caller-codec-list PCMU
  callee-codec-list PCMA
  media police strip
  action cac-complete
  complete
  cac-policy-set global 1
codec-list PCMU
codec PCMU
exit
codec-list PCMA
codec PCMA
exit
end
```

トランスレーティングをイネーブルにするための Unified SBC の設定例



(注)

トランスレーティングは、コーデック間ではなく、同じコーデックを使用した異なるレートに対してのみサポートされます。したがって、トランスレーティングとトランスコーディングは同時に実行できません。

次に、同じコーデック ポリシーを使用してトランスレーティング機能をイネーブルにするために Unified SBC を設定する例を示します。

```
enable
configure terminal
sbc mySBC
sbe
cac-policy-set 2
first-cac-table Transrate
cac-table Transrate
table-type policy-set
entry 1
cac-scope call
callee ptime 30
caller ptime 20
media police strip
action cac complete
complete
cac-policy-set global 2
end
```

次に、同じコーデック ポリシーを使用してトランスレーティング機能をイネーブルにするために Unified SBC を設定する例を示します。

```
enable
configure terminal
sbc MySBC
sbe
codec list PCMU
policy transrating
codec PCMU packetization-period 200 priority 1
end
```

SRTP とトランスコーディングをイネーブルにするための Unified SBC の設定例

次に、SRTP とトランスコーディング機能をイネーブルにするように SBC を設定する例を示します。

```
enable
configure terminal
sbc mySBC
sbe
cac-policy-set 3
first-cac-table C3
cac-table c3
table-type policy-set
entry 1
cac-scope global
srtp support allow
srtp caller mandate
srtp callee mandate
srtp interworking allow
srtp media interworking allow
action next-table xcode
exit
cac-table xcode
table-type limit dst-adjacency
entry 1
match-value nav4b
callee-codec-list PCMU
```



```
action cac-complete
complete
cac-policy-set global 3
end
```

インバンド DTMF インターワーキングのための Unified SBC の設定例

次に、Unified SBC でインバンド DTMF 送信を設定する例を示します。



(注)

この手順では、**caller** コマンドと **callee** コマンドが使用されています。シナリオによっては、**caller** と **callee** のコマンド ペアの代わりに **branch** コマンドを使用できます。**branch** コマンドはリリース 3.5.0 で導入されました。このコマンドの詳細については、「[ダイレクト非制限 CAC ポリシーの設定](#)」(P.139) を参照してください。

```
enable
configure terminal
sbc mySBC
sbe
cac-policy-set 4
first-cac-table c4
cac-table c4
table-type policy-set
entry 1
cac-scope global
callee inband-dtmf-mode always
caller inband-dtmf-mode never
action next-table xcode
exit
exit
cac-table xcode
table-type limit dst-adjacency
entry 1
match-value spab
callee-codec-list PCMU
action cac-complete
complete
cac-policy-set global 4
end
```

