CHAPTER 18

SDP Bandwidth Field Features

Cisco Unified Border Element (SP Edition) supports the Bandwidth Field Interworking and the Option to Use CODEC Instead of Bandwidth-Field for Media Bandwidth Allocation features in the unified model.

In a Session Initiation Protocol (SIP) message exchange, Cisco Unified Border Element (SP Edition) uses the parameters defined in the Session Description Protocol (SDP) bandwidth-fields (b-line) for calculating the media pinhole bandwidth. During SIP message exchange, the SDP may contain both the bandwidth-fields and coder/decoder (CODEC) information. In that case, Cisco Unified Border Element (SP Edition) would use the bandwidth-field value to allocate sufficient bandwidth for the media pinhole.

During deployment, there might be some endpoints for which it would be better to set the media pinhole bandwidth using a CODEC definition in the Session Description Protocol (SDP) messages instead of using b-line.

The Option to Use CODEC Instead of Bandwidth-Field for Media Bandwidth Allocation feature allows you to set a media bandwidth flag in a Call Admission Control (CAC) policy entry to ignore the b-line and use CODEC for calculating the media pinhole bandwidth.

Cisco Unified Border Element (SP Edition) supports Bandwidth Field Interworking by supporting the ability to determine how bandwidth lines are translated in the outbound Session Description Protocol (SDP) sent to the caller and callee with bandwidth line passthrough using Application Specific Maximum (AS) and Transport Independent Application Specific Maximum (TIAS) conversion.

Note For Cisco IOS XE Release 2.4 and later, these features are supported in the unified model only.

Cisco Unified Border Element (SP Edition) was formerly known as Integrated Session Border Controller and may be commonly referred to in this document as the session border controller (SBC).

For a complete description of the commands used in this chapter, refer to the *Cisco Unified Border Element (SP Edition) Command Reference: Unified Model* at:

http://www.cisco.com/en/US/docs/ios/sbc/command/reference/sbcu_book.html.

For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or a Cisco IOS master commands list.

Feature History for SDP Bandwidth Field Features

Release	Modification
Cisco IOS XE Release 2.4	Option to Use CODEC Instead of Bandwidth-Field for Media Bandwidth Allocation feature was introduced on the Cisco IOS XR, along with the unified model.

Cisco IOS XE Release 2.5	Bandwidth Field Interworking feature was introduced on the Cisco IOS XR.
Cisco IOS XE Release 3.1S	Media Bandwidth Policy feature was added.
Cisco IOS XE Release 3.2S	Per-adjacency codec string interworking feature was added.

Contents

This chapter contains the following sections:

Prerequisites for Implementing SDP Bandwidth Field Features

The following prerequisite is required to implement SDP Bandwidth Field features:

Before implementing SDP Bandwidth Field features, Cisco Unified Border Element (SP Edition) must already be configured.

Option to Use CODEC Instead of Bandwidth-Field for Media Bandwidth Allocation Feature

The following sections are in the “Option to Use CODEC Instead of Bandwidth-Field for Media Bandwidth Allocation” feature:

- Information About Calculating Bandwidth in SIP Calls, page 18-3
- Configuring Option to Use CODEC Instead of Bandwidth-Field for Media Bandwidth Allocation, page 18-3
- Option to Use CODEC Instead of Bandwidth-Field for Media Bandwidth Allocation Configuration: Example, page 18-7

Information About Calculating Bandwidth in SIP Calls

The SBC analyzes each media stream in a call and calculates the bandwidth required. For SIP calls containing SDP, the SBC looks for any b=CT, b=AS, or b=TIAS lines. If present, these lines are used to calculate the baseline bandwidth required for the media stream.

If these lines are not present, the SBC calculates the baseline bandwidth by inspecting each of the possible CODECs in the stream and calculating the baseline bandwidth based on them. The bandwidth allocated per CODEC is configurable.

The SBC then adjusts the baseline bandwidth to take into account any necessary packetization and Real Time Control Protocol (RTCP) bandwidth overheads.

Certain endpoints do not conform to the bandwidth requirements that the SBC calculates for a media stream, for example:

- Endpoints that start renegotiating the bandwidth for a call can start using additional bandwidth before the renegotiation is complete.
- Endpoints that request an incorrect bandwidth for secure media using the b-line, because they do not take into account the increased payload size required for the encryption.
- Endpoints that transmit data in multiple formats in parallel (such as high and low definition video using different payloads in a single stream) without taking into account all formats when calculating the bandwidth requirements.

To allow interoperation with these endpoints without dropping packets, the SBC allows the per-CODEC bandwidth to be configurable. This allows the SBC administrator to set a suitably large maximum value for CODECs supported by these endpoints. This is sufficient because the endpoints in question use a well known set of CODECs.

However, if the endpoint includes an explicit bandwidth (b-) line, then the SBC uses that to calculate the bandwidth instead of the maximum value. The Option to Use CODEC Instead of Bandwidth-Field for Media Bandwidth Allocation feature uses the **media bandwidth-fields ignore** command to set a media flag in a Call Admission Control (CAC) policy entry to ignore the b-line and use CODEC to calculate the bandwidth.

Configuring Option to Use CODEC Instead of Bandwidth-Field for Media Bandwidth Allocation

This task configures the Option to Use CODEC Instead of Bandwidth-Field for Media Bandwidth Allocation feature.

SUMMARY STEPS

1. **configure terminal**
2. **sbc** *service-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. **media bandwidth-fields ignore**
11. **action** [next-table *goto-table-name* | cac-complete]
12. **exit**
13. **exit**
14. **complete**

DETAILED STEPS

	Command or Action	Purpose

Step 1	`configure terminal`
	Example:
	`Router# configure`
		Enables global configuration mode.

Step 2	`sbc` *`service-name`*
	Example:
	`Router(config)# sbc mysbc`
		Enters the mode of an SBC service.
		• Use the *service-name* argument to define the name of the service.

Step 3	`sbe`
	Example:
	`Router(config-sbc)# sbe`
		Enters the mode of an SBE entity within an SBC service.

Step 4	`cac-policy-set` *`policy-set-id`*
	Example:
	`Router(config-sbc-sbe)# cac-policy-set 1`
		Enters the mode of CAC policy set configuration within an SBE entity, creating a new policy set if necessary.

Step 5	`first-cac-table` *`table-name`*
	Example:
	`Router(config-sbc-sbe-cacpolicy)# first-cac-table StandardListByAccount`
		Configures the name of the first policy table to process when performing the admission control stage of policy.

Step 6	`cac-table` *`table-name`*
	Example:
	`Router(config-sbc-sbe-cacpolicy)# cac-table StandardListByAccount`
		Enters the mode for configuration of an admission control table (creating one if necessary) within the context of an SBE policy set.

	Command or Action	Purpose
Step 7	**table-type {policy-set | limit** {*list of limit tables*}} **Example:** `Router(config-sbc-sbe-cacpolicy-cactable)# table-type policy-set`	Configures the table type of a CAC table within the context of an SBC policy set. The *list of limit tables* argument controls the syntax of the match-value fields of the entries in the table. Possible available Limit tables are: • *account*—Compare the name of the account. • *adj-group*—Compare the name of the adjacency group. • *adjacency*—Compare the name of the adjacency. • *all*—No comparison type. All events match this type. • *call-priority*—Compare with call priority. • *category*—Compare the number analysis assigned category. • *dst-account*—Compare the name of the destination account. • *dst-adj-group*—Compare the name of the destination adjacency group. • *dst-adjacency*—Compare the name of the destination adjacency. • *dst-prefix*—Compare the beginning of the dialed digit string. • *event-type*—Compare with CAC policy event types. • *src-account*—Compare the name of the source account. • *src-adj-group*—Compare the name of the source adjacency group. • *src-adjacency*—Compare the name of the source adjacency. • *src-prefix*—Compare the beginning of the calling number string. Features can be enabled or disabled per adjacency group through CAC configuration the same way this is done per individual adjacencies. The adj-group table type matches on either source or destination adjacency group. When the policy-set keyword is specified, use the **cac-scope** command to configure the scope within each entry at which limits are applied in a CAC Policy Set table.
Step 8	**entry** *entry-id* **Example:** `Router(config-sbc-sbe-cacpolicy-cactable)# entry 1`	Enters the mode to create or modify an entry in an admission control table.

	Command or Action	Purpose
Step 9	**cac-scope {***list of scope options***}** **Example:** `Router(config-sbc-sbe-cacpolicy-cactable-entry) # cac-scope src-adjacency`	Choose a scope at which CAC limits are applied within each of the entries in a policy set table. • *list of scope options*—Specifies one of the following strings used to match events: – **account**—Events that are from the same account. – **adjacency**—Events that are from the same adjacency. – **adj-group**—Events that are from members of the same adjacency group. – **call**—Scope limits are per single call. – **category**—Events that have same category. – **dst-account**—Events that are sent to the same account. – **dst-adj-group**—Events that are sent to the same adjacency group. – **dst-adjacency**—Events that are sent to the same adjacency. – **dst-number**—Events that have same destination. – **global**—Scope limits are global – **src-account**—Events that are from the same account. – **src-adj-group**—Events that are from the same adjacency group. – **src-adjacency**—Events that are from the same adjacency. – **src-number**—Events that have the same source number. – **sub-category**—The limits specified in this scope apply to all events sent to or received from members of the same subscriber category. – **sub-category-pfx**—The limits specified in this scope apply to all events sent to or received from members of the same subscriber category prefix. – **subscriber**—The limits specified in this scope apply to all events sent to or received from individual subscribers (a device that is registered with a Registrar server).
Step 10	**media bandwidth-fields ignore** **Example:** `Router(config-sbc-sbe-cacpolicy-cactable-entry) # media bandwidth-fields ignore`	Sets the media flag to ignore the b-line and use CODEC to calculate the baseline bandwidth required for the media stream.

	Command or Action	Purpose

Step 11	`action [next-table goto-table-name | cac-complete]`
	Example:
	`Router(config-sbc-sbe-cacpolicy-cactable-entry) # action cac-complete`
		Configures the action to perform after this entry in an admission control table. Possible actions are:
		• Identify the next CAC table to process using the **next-table** keyword and the *goto-table-name argument.*
		• Stop processing for this scope using the **cac-complete** keyword.

Step 12	`exit`
	Example:
	`Router(config-sbc-sbe-cacpolicy-cactable-entry) # exit`
		Exits from **entry** to **cactable** mode.

Step 13	`exit`
	Example:
	`Router(config-sbc-sbe-cacpolicy-cactable)# exit`
		Exits from **cactable** to **cacpolicy** mode.

Step 14	`complete`
	Example:
	`Router(config-sbc-sbe-cacpolicy)# complete`
		Completes the CAC policy set when you have committed the full set.

Option to Use CODEC Instead of Bandwidth-Field for Media Bandwidth Allocation Configuration: Example

The following example shows how to set the media flag to ignore the b-line and use CODEC to calculate the baseline bandwidth required for the media stream:

```
Router# configure terminal
Router(config)# sbc mysbc
Router(config-sbc)# sbe
Router(config-sbc-sbe)# cac-policy-set 1
Router(config-sbc-sbe-cacpolicy)# first-cac-table StandardListByAccount
Router(config-sbc-sbe-cacpolicy)# cac-table StandardListByAccount
Router(config-sbc-sbe-cacpolicy-cactable)# table-type policy-set
Router(config-sbc-sbe-cacpolicy-cactable)# entry 1
Router(config-sbc-sbe-cacpolicy-cactable-entry)# cac-scope src-adjacency
Router(config-sbc-sbe-cacpolicy-cactable-entry)# media bandwidth-fields ignore
Router(config-sbc-sbe-cacpolicy-cactable-entry)# action cac-complete
Router(config-sbc-sbe-cacpolicy-cactable-entry)# exit
Router(config-sbc-sbe-cacpolicy-cactable)# exit
Router(config-sbc-sbe-cacpolicy-cactable-entry)# complete
```

Information About Media Bandwidth Policy

Previous to this release, SBC disabled or rejected media calls that exceeded the bandwidth allowed by the Call Admission Control (CAC). However, some applications, such as Telepresence, require other options. SBC now provides the ability to diminish the video stream to a lower bandwidth, while allowing the audio stream to remain unchanged.

In this release, bandwidth restrictions are enhanced by allowing the user to configure one of three media bandwidth options, using the **media police** command:

- strip
- reject
- degrade

These options can be configured for all media types or for video only if desired.

strip

If an individual media stream exceeds the bandwidth limit for a call, that media stream is disabled by setting the port to zero (0). If after the above stage has completed, the sum of the bandwidths of all remaining streams exceeds the bandwidth limit for a call, the request is rejected.

When the port is set to zero (0), the call is ended and the following message is displayed on the screen:

```
incompatible sites
```

reject

If an individual media stream exceeds the bandwidth limit for a call, the request is rejected. If the sum of the bandwidths of all media streams exceeds the bandwidth limit for a call, the request is rejected.

degrade

If a media stream exceeds the bandwidth limit for a call, the video stream is downgraded to a lower (non-zero) bandwidth that brings the media stream within the bandwidth limit for the call.

Note Only the video stream is downgraded. Audio streams are not downgraded. If the audio stream exceeds the bandwidth for a call, the media stream cannot be downgraded.

Restrictions

The degrade option is not supported on H.323 calls.

Using the degrade option may cause a 2 to 5 percent performance degradation.

Configuration

You configure the Media Bandwidth Policy by configuring the media policy mode, using the **media policy** command, and by configuring the minimum bandwidth for the for the analog-to-digital codec (enCOder/DECoder) hardware, using the **bandwidth** command.

Note The codec name must be one of the system codecs that SBC can recognize. To see a list of the system codecs, use the **show sbc sbc sbe codecs** command.

The **max-bandwidth-per-scope** command specifies the maximum bandwidth limit for all media streams in all directions, including packet overheads.

The **bandwidth min** command specifies the unidirectional, minimum bandwidth limit bandwidth and does not include packet overhead.

See the ?$paranum>Configuring Media Bandwidth Policy? section on page 18-9 for the configuration steps and the ?$paranum>end? section on page 18-30 for configuration examples.

Configuring Media Bandwidth Policy

This section provides the following step procedures:

- Configuring the Media Policy Mode, page 18-9
- Configuring the Codec Minimum Bandwidth, page 18-10
- Configuring the Maximum Bandwidth Per Scope, page 18-12

Configuring the Media Policy Mode

Use the following procedure to configure the media policy mode.

SUMMARY STEPS

1. **config**
2. **sbc** *sbc-name*
3. **sbe**
4. **cac-policy-set** *policy-set-id*
5. **cac-table** *table-name*
6. **table-type** policy-set
7. **entry** *entry-id*
8. **media police strip** | **reject** | **degrade**
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**config** **Example:** Router# config	Enters global configuration mode.
Step 2	**sbc** ***sbc-name*** **Example:** Router(config)# sbc SBC1	Creates the SBC service on Cisco Unified Border Element (SP Edition) and enters into SBC configuration mode.
Step 3	**sbe** **Example:** Router(config-sbc)# sbe	Enters the mode of the signaling border element (SBE) function of the SBC.

	Command or Action	Purpose
Step 4	**cac-policy-set** *policy-set-id* **Example:** `Router(config-sbc-sbe)# cac-policy-set 1`	Enters the mode of CAC policy set configuration within an SBE entity, creating a new policy set if necessary. *policy-set-id*—Integer chosen by the user to identify the policy set. The range is 1 to 2147483647.
Step 5	**cac-table** *table-name* **Example:** `Router(config-sbc-sbe-cacpolicy)# cac-table testSecure`	Enters the mode for configuration of an admission control table (creating one if necessary) within the context of an SBE policy set. *table-name*—Name of the admission control table.
Step 6	**table-type policy-set** **Example:** `Router(config-sbc-sbe-cacpolicy-cactable)# table-type policy-set`	Configures the table type of a CAC table within the context of an SBE policy set.
Step 7	**entry** *entry-id* **Example:** `Router(config-sbc-sbe-cacpolicy-cactable)# entry 1`	Enters the mode to modify an entry in an admission control table. *entry-id*—Specifies the table entry.
Step 8	**media police strip | reject | degrade** **Example:** `Router(config-sbc-sbe-cacpolicy-cactable-entry) # media police degrade`	Configures how SBC handles media streams that exceed bandwidth limits for media calls.

Step 9	**end**
	Example:
	`Router(config-sbc-sbe-cacpolicy-cactable-entry) # end`
		Exits configuration mode and returns to privileged EXEC mode.

Configuring the Codec Minimum Bandwidth

Use the following procedure to configure the minimum bandwidth for the codec.

SUMMARY STEPS

1. **config**
2. **sbc** *sbc-name*
3. **sbe**
4. **codec custom** *custom-name* **id**
5. **type variable**
6. **media video**
7. **bandwidth min** *bandwidth-value*
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Router# config`	Enters global configuration mode.
Step 2	**sbc** *sbc-name* **Example:** `Router(config)# sbc SBC1`	Creates the SBC service on Cisco Unified Border Element (SP Edition) and enters into SBC configuration mode.
Step 3	**sbe** **Example:** `Router(config-sbc)# sbe`	Enters the mode of the signaling border element (SBE) function of the SBC.
Step 4	**codec custom** *custom-name* **id** **Example:** `Router (config-sbc-sbe)# codec custom h263-c id 96`	Specifies the name of the custom analog-to-digital codec (enCOder/DECoder) and enters codec definition mode.
Step 5	**type variable** **Example:** `Router (config-sbc-sbe-codec-def)# type variable`	Sets the type of the codec to variable.
Step 6	**media video** **Example:** `Router (config-sbc-sbe-codec-def)# media video`	Sets the media type to video.
Step 7	**bandwidth min** *bandwidth-value* **Example:** `Router (config-sbc-sbe-codec-def)# bandwidth min 328000`	Sets the minimum bandwidth for the codec. **Note** The **bandwidth min** command specifies the unidirectional, minimum bandwidth limit and does not include packet overhead.

Step 8	**end**
	Example:
	`Router(config-sbc-sbe-cacpolicy-cactable-entry)# end`
		Exits configuration mode and returns to privileged EXEC mode.

Configuring the Maximum Bandwidth Per Scope

Use the following procedure to configure the bandwidth limit for all media streams.

SUMMARY STEPS

1. **configure terminal**
2. **sbc** *sbc-name*
3. **sbe**
4. **cac-policy-set** *policy-set-id*
5. **description** *description*
6. **first-cac-table** *table-name*
7. **first-cac-scope** *scope-name*
8. **cac-table** *table-name*
9. **table-type policy-set**
10. **entry** *entry-id*
11. **max-bandwidth-per-scope** *bandwidth*
12. **action cac-complete**
13. **media police degrade**
14. **complete**
15. **codec system** *sys-codec* **id** *payload id*
16. **type variable**
17. **bandwidth min** *bandwidth-value*
18. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Router# config t`	Enters global configuration mode.
Step 2	**sbc** *sbc-name* **Example:** `Router(config)# sbc SBC1`	Creates the SBC service on Cisco Unified Border Element (SP Edition) and enters into SBC configuration mode.
Step 3	**sbe** **Example:** `Router(config-sbc)# sbe`	Enters the mode of the signaling border element (SBE) function of the SBC.

	Command or Action	Purpose
Step 4	**cac-policy-set** *policy-set-id* **Example:** `Router(config-sbc-sbe)# cac-policy-set 1`	Enters the mode of CAC policy set configuration within an SBE entity, creating a new policy set if necessary. *policy-set-id*—Integer chosen by the user to identify the policy set. The range is 1 to 2147483647.
Step 5	**description** *description* **Example:** `Router(config-sbc-sbe-cacpolicy)# description bandwidth degrade`	Configures descriptive text for this policy set.
Step 6	**first-cac-table** *table-name* **Example:** `Router(config-sbc-sbe-cacpolicy)# first-cac-table my_table`	Configures the name of the first policy table to process. A CAC policy may have many tables configured. To start the application of the CAC policy, the first table that is used needs to be defined. *table-name*—The admission control table that should be processed first.
Step 7	**first-cac-scope** *scope-name* **Example:** `Router(config-sbc-sbe-cacpolicy)# first-cac-scope call`	Configures scope at which limits should be initially defined when performing the admission control stage of the policy. Each CAC policy has a scope that is applied to it. This CAC policy applies on a per call basis. *scope-name* has one of the following values: • **adj-group**—Limits for events from members of the same adjacency group. • **call**—Limits are per single call. • category—Limits per category. • **dst-account**—Limits for events sent to the same account. • **dst-adj-group**—Limits for events sent to the same adjacency group. • **dst-adjacency**—Limits for events sent to the same adjacency. • **dst-number**—Limits for events that have the same adjacency number. • **global**—Limits are global (May not be combined with any other option). • **src-account**—Limits for events from the same account. • **src-adj-group**—Limits for events from the same adjacency group. • **arc-adjacency**—Limits for events from the same adjacency. • **src-number**—Limits for events that have the same source number.

	Command or Action	Purpose
Step 8	**cac-table** *table-name* **Example:** Router(config-sbc-sbe-cacpolicy)# cac-table testSecure	Enters the mode for configuration of an admission control table (creating one if necessary) within the context of an SBE policy set. *table-name*—Name of the admission control table.
Step 9	**table-type {policy-set | limit** {*list of limit tables*}} **Example:** Router(config-sbc-sbe-cacpolicy-cactable)# table-type policy-set	Configures the table type of a CAC table within the context of an SBE policy set. For Policy Set tables, the event or call or message is applied to all entries in this table.
Step 10	**entry** *entry-id* **Example:** Router(config-sbc-sbe-cacpolicy-cactable)# entry 1	Enters the mode to modify an entry in an admission control table. *entry-id*—Specifies the table entry.
Step 11	**max-bandwidth-per-scope** *bandwidth* **Example:** Router(config-sbc-sbe-cacpolicy-cactable-entry) # max-bandwidth-per-scope 6000000 bps	Configures the maximum limit for the bandwidth in bps, Kbps, Mbps or Gbps for an entry in an admission control table. **Note** The **max-bandwidth-per-scope** command specifies the maximum bandwidth limit for all media streams in all directions, including packet overheads.
Step 12	**action cac-complete** **Example:** Router(config-sbc-sbe-cacpolicy-cactable-entry) # action cac-complete	Configures the action to perform after this entry in an admission control table. In this case, stop processing for this scope using the **cac-complete** keyword.
Step 13	**media police strip** | **reject** | **degrade** **Example:** Router(config-sbc-sbe-cacpolicy-cactable-entry) # media police degrade	Configures how SBC handles media streams that exceed bandwidth limits for media calls. In this case, **degrade**.
Step 14	**complete** **Example:** Router(config-sbc-sbe-cacpolicy-cactable-entry) complete	Completes the CAC-policy or call-policy set after committing the full set.
Step 15	**codec system** *sys-codec* **id** *payload id* **Example:** Router(config-sbc-sbe-cacpolicy)# codec system H263 id 34	Enters codec definition mode to modify an existing codec.

	Command or Action	Purpose
Step 16	**type variable** **Example:** `Router (config-sbc-sbe-codec-def)# type variable`	Sets the type of the codec to variable.
Step 17	**bandwidth min** *bandwidth-value* **Example:** `Router (config-sbc-sbe-codec-def)# bandwidth min 370000`	Sets the minimum bandwidth for the codec. **Note** The **bandwidth min** command specifies the unidirectional, minimum bandwidth limit bandwidth and does not include packet overhead.

Step 18	**end**
	Example:
	`Router(config-sbc-sbe-cacpolicy-cactable-entry) # end`
		Exits configuration mode and returns to privileged EXEC mode.

Media Policy Configuration: Examples

This section provides the following configuration examples:

- Media Policy Mode Configuration: Example, page 18-16
- Codec Minimum Bandwidth Configuration: Example, page 18-16
- Maximum Bandwidth Per Scope Configuration: Example, page 18-16

Media Policy Mode Configuration: Example

The following example shows how to configure SBC to degrade media streams to lower bandwidths when requests exceed bandwidth limits.

```
Router# config t
Router(config)# sbc mySBC
Router(config-sbc)# sbe
Router(config-sbc-sbe)# cac-policy-set 1
Router(config-sbc-sbe-cacpolicy)# cac-table cac-tbl-1
Router(config-sbc-sbe-cacpolicy-cactable)# table-type policy-set
Router(config-sbc-sbe-cacpolicy-cactable)# entry 1
Router(config-sbc-sbe-cacpolicy-cactable-entry)# media police degrade
Router(config-sbc-sbe-cacpolicy-cactable-entry)#
```

Codec Minimum Bandwidth Configuration: Example

The following example shows how to configure the maximum bandwidth limit to 400,000 bps for media calls:

```
Router# config t
Router(config)# sbc mySBC
Router(config-sbc)# sbe
Router(config-sbc-sbe)# codec system H263 id 34
Router(config-sbc-sbe-codec-def)# bandwidth 400000
```

The following example shows how to configure the minimum bandwidth limit to 328,000 bps, specifically for video type media calls:

```
Router# config t
Router(config)# sbc mySBC
Router(config-sbc)# sbe
Router (config-sbc-sbe)# codec custom h263-c id 96
Router (config-sbc-sbe-codec-def)# type variable
Router (config-sbc-sbe-codec-def)# media video
Router (config-sbc-sbe-codec-def)# bandwidth min 328000
```

Maximum Bandwidth Per Scope Configuration: Example

The following example shows how to configure the bandwidth limit for all media streams:

```
Router# config t
Router(config)# sbc SBC1
Router(config-sbc)# sbe
Router(config-sbc-sbe)# cac-policy-set 1
Router(config-sbc-sbe-cacpolicy)# description bandwidth degrade
Router(config-sbc-sbe-cacpolicy)# first-cac-table my_table
Router(config-sbc-sbe-cacpolicy)# first-cac-scope call
Router(config-sbc-sbe-cacpolicy)# cac-table testSecure
Router(config-sbc-sbe-cacpolicy-cactable)# table-type policy-set
Router(config-sbc-sbe-cacpolicy-cactable)# entry 1
Router(config-sbc-sbe-cacpolicy-cactable-entry)# max-bandwidth-per-scope 6000000 bps
Router(config-sbc-sbe-cacpolicy-cactable-entry)# action cac-complete
Router(config-sbc-sbe-cacpolicy-cactable-entry)# media police degrade
Router(config-sbc-sbe-cacpolicy-cactable-entry) complete
Router(config-sbc-sbe-cacpolicy)# codec system H263 id 34
Router (config-sbc-sbe-codec-def)# type variable
Router (config-sbc-sbe-codec-def)# bandwidth min 370000
Router(config-sbc-sbe-cacpolicy-cactable-entry)# end
```

Bandwidth Field Interworking Feature

The following sections are in the Bandwidth Field Interworking feature:

- Information About Bandwidth Field Interworking, page 18-17
- Configuring Bandwidth Field Interworking, page 18-18
- Bandwidth Field Interworking Configuration: Examples, page 18-22

Information About Bandwidth Field Interworking

Cisco Unified Border Element (SP Edition) supports Bandwidth Field Interworking by supporting the ability to determine how bandwidth lines are configured in the outbound Session Description Protocol (SDP). Cisco Unified Border Element (SP Edition) supports the Application Specific Maximum (AS) and Transport Independent Application Specific Maximum (TIAS) bandwidth modifiers in the SDP.

The SDP includes an optional bandwidth attribute with the following syntax, according to RFC 3556, Session Description Protocol (SDP) Bandwidth Modifiers:

b=<modifier>:<bandwidth-value>

The <modifier> is an alphanumeric word that indicates the bandwidth to be used by the media or session. The <bandwidth- value> default is kilobits per second.

The AS bandwidth modifier is used to specify the total bandwidth for a single media stream from one source.

The TIAS bandwidth value is the maximum bandwidth required by the SDP session level or media stream without counting IP or other transport layers like TCP or UDP (RFC 3890).

Cisco Unified Border Element (SP Edition) supports translation between the AS and TIAS bandwidth formats which are configured for each adjacency by means of the following commands:

- **caller-bandwidth-field** *[as-to-tias* | *tias-to-as]*
- **callee-bandwidth-field** *[as-to-tias* | *tias-to-as]*

When you configure the bandwidth line to the as-to-tias setting, this causes the SBC, in an outbound SDP offer, to convert a b=AS line into a b=TIAS line. If there are multiple b=AS lines, only the first line is converted into a b=TIAS line and the rest are ignored.

Translating from a bandwidth modifier of AS into TIAS can be useful in the following situations:

- If operating with upstream devices that only support the AS bandwidth modifier, use of the TIAS bandwidth modifier downstream may improve the accuracy of bandwidth calculations in the network. In some network scenarios, use of the AS bandwidth modifier may lead to incorrect bandwidth calculations, for example, if routing between an IPv4 and IPv6 network (see RFC3890).

- For interoperability purposes—if there are downstream devices that do not understand the AS bandwidth modifier.

When you configure the bandwidth line to the tias-to-as setting, this causes the SBC, in an outbound SDP offer, to convert a b=TIAS line into a b=AS line if there is not already a b=AS line associated with that SDP media descriptor. If there are multiple b=TIAS lines, only the first is converted into a b=AS line and the rest are ignored.

Translating from a bandwidth modifier of TIAS into AS can be useful in the following situation:

- For interoperability purposes—if there are downstream nodes that do not understand the TIAS bandwidth modifier.

The SBC supports translation between these two formats. If bandwidth line conversion is enabled for the offerer-side adjacency, then an answer has its bandwidth lines converted to the specified format before being sent back to the offerer. Similarly, If bandwidth line conversion is enabled for the answer-side adjacency, then an offer has its bandwidth lines converted to the specified format before being sent to the answer.

The same rules are applied to translation of bandwidth lines in the answer and translation of bandwidth lines in the offer. The rules are as follows:

- The SBC conforms to whichever outgoing bandwidth line format is configured. If the outgoing adjacency is configured to prefer a specific style of bandwidth line format, then that format is used. Thus any AS or TIAS bandwidth lines are translated to that format.
- If the offerer's adjacency has no configured bandwidth line format preference, but a translation in bandwidth line format was made on the offer to accommodate the answer-side adjacency's preference, then the reverse translation is done on the answer.

  For example, the answer adjacency is configured to translate to TIAS bandwidth lines. The offerer's adjacency has no preference. The offerer makes an SDP offer containing b=AS lines, which are then converted by the SBC to the b=TIAS style for the outgoing offer.

  The answerer responds with a b=TIAS line which represents an increased bandwidth requirement. This increased bandwidth line is translated back to b=AS before being sent to the offerer because that is the style the offerer last offered with.

Configuring Bandwidth Field Interworking

This task configures the Bandwidth Field Interworking feature.

Note The **caller** and **callee** commands have been used in this procedure. In some scenarios, the **branch** command can be used as an alternative to the **caller** and **callee** command pair. The **branch** command has been introduced in Release 3.5.0. See the ?$paranum>Configuring Directed Nonlimiting CAC Policies? section on page 7-37 for information about this command.

SUMMARY STEPS

1. **configure**
2. **sbc** *service-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. **caller-bandwidth-field [as-to-tias] [tias-to-as]**
11. **callee-bandwidth-field [as-to-tias] [tias-to-as]**
12. **action** [next-table *goto-table-name* | cac-complete]
13. **exit**
14. **exit**
15. **complete**

DETAILED STEPS

	Command or Action	Purpose

Step 1	`configure terminal`
	Example:
	`Router# configure`
		Enables global configuration mode.

Step 2	**`sbc`** *`service-name`*
	Example:
	`Router(config)# sbc mysbc`
		Enters the mode of an SBC service.
		• Use the *service-name* argument to define the name of the service.

Step 3	**`sbe`**
	Example:
	`Router(config-sbc)# sbe`
		Enters the mode of an SBE entity within an SBC service.

Step 4	**`cac-policy-set`** *`policy-set-id`*
	Example:
	`Router(config-sbc-sbe)# cac-policy-set 1`
		Enters the mode of CAC policy set configuration within an SBE entity, creating a new policy set if necessary.

Step 5	**`first-cac-table`** *`table-name`*
	Example:
	`Router(config-sbc-sbe-cacpolicy)# first-cac-table StandardListByAccount`
		Configures the name of the first policy table to process when performing the admission control stage of policy.

Step 6	**`cac-table`** *`table-name`*
	Example:
	`Router(config-sbc-sbe-cacpolicy)# cac-table StandardListByAccount`
		Enters the mode for configuration of an admission control table (creating one if necessary) within the context of an SBE policy set.

	Command or Action	Purpose
Step 7	**table-type {policy-set | limit** {*list of limit tables*}} **Example:** `Router(config-sbc-sbe-cacpolicy-cactable)# table-type policy-set`	Configures the table type of a CAC table within the context of an SBC policy set. The *list of limit tables* argument controls the syntax of the match-value fields of the entries in the table. Possible available Limit tables are: • *account*—Compare the name of the account. • *adj-group*—Compare the name of the adjacency group. • *adjacency*—Compare the name of the adjacency. • *all*—No comparison type. All events match this type. • *call-priority*—Compare with call priority. • *category*—Compare the number analysis assigned category. • *dst-account*—Compare the name of the destination account. • *dst-adj-group*—Compare the name of the destination adjacency group. • *dst-adjacency*—Compare the name of the destination adjacency. • *dst-prefix*—Compare the beginning of the dialed digit string. • *event-type*—Compare with CAC policy event types. • *src-account*—Compare the name of the source account. • *src-adj-group*—Compare the name of the source adjacency group. • *src-adjacency*—Compare the name of the source adjacency. • *src-prefix*—Compare the beginning of the calling number string. Features can be enabled or disabled per adjacency group through CAC configuration the same way this is done per individual adjacencies. The adj-group table type matches on either source or destination adjacency group. When the policy-set keyword is specified, use the **cac-scope** command to configure the scope within each entry at which limits are applied in a CAC Policy Set table.
Step 8	**entry** *entry-id* **Example:** `Router(config-sbc-sbe-cacpolicy-cactable)# entry 1`	Enters the mode to create or modify an entry in an admission control table.

	Command or Action	Purpose
Step 9	**cac-scope {***list of scope options***}** **Example:** `Router(config-sbc-sbe-cacpolicy-cactable-entry) # cac-scope src-adjacency`	Choose a scope at which CAC limits are applied within each entry in a Policy Set table. *list of scope options*—Specifies one of the following strings used to match events: • *account*—Events that are from the same account. • *adjacency*—Events that are from the same adjacency. • *adj-group*—Events that are from members of the same adjacency group. • *call*—Scope limits are per single call. • *category*—Events that have same category. • *dst-account*—Events that are sent to the same account. • *dst-adj-group*—Events that are sent to the same adjacency group. • *dst-adjacency*—Events that are sent to the same adjacency. • *dst-number*—Events that have the same destination. • *global*—Scope limits are global • *src-account*—Events that are from the same account. • *src-adj-group*—Events that are from the same adjacency group. • *src-adjacency*—Events that are from the same adjacency. • *src-number*—Events that have the same source number.

Step 10	**caller-bandwidth-field [as-to-tias] [tias-to-as]**
	Example:
	`Router(config-sbc-sbe-cacpolicy-cactable-entry) # caller-bandwidth-field as-to-tias`
		Configures the SBC to convert a b=AS line format into a b=TIAS line format or a b=TIAS line format into a b=AS line format in an outbound Session Description Protocol (SDP) sent to the caller.
		AS = Application Specific Maximum
		TIAS = Transport Independent Application Specific Maximum

Step 11	**callee-bandwidth-field [as-to-tias] [tias-to-as]**
	Example:
	`Router(config-sbc-sbe-cacpolicy-cactable-entry) # callee-bandwidth-field tias-to-as`
		Configures the SBC to convert a b=AS line format into a b=TIAS line format or a b=TIAS line format into a b=AS line format in an outbound Session Description Protocol (SDP) sent to the callee.
		AS = Application Specific Maximum
		TIAS = Transport Independent Application Specific Maximum

	Command or Action	Purpose
Step 12	**action** [**next-table** *goto-table-name* | **cac-complete**] **Example:** `Router(config-sbc-sbe-cacpolicy-cactable-entry) # action cac-complete`	Configures the action to perform after this entry in an admission control table. Possible actions are: • Identify the next CAC table to process using the **next-table** keyword and the *goto-table-name argument.* • Stop processing for this scope using the **cac-complete** keyword.
Step 13	**exit** **Example:** `Router(config-sbc-sbe-cacpolicy-cactable-entry) # exit`	Exits from **entry** to **cactable** mode.
Step 14	**exit** **Example:** `Router(config-sbc-sbe-cacpolicy-cactable)# exit`	Exits from **cactable** to **cacpolicy** mode.
Step 15	**complete** **Example:** `Router(config-sbc-sbe-cacpolicy-cactable-entry) # complete`	Completes the CAC policy set when you have committed the full set.

Bandwidth Field Interworking Configuration: Examples

Note The **caller** and **callee** commands have been used in this procedure. In some scenarios, the **branch** command can be used as an alternative to the **caller** and **callee** command pair. The **branch** command has been introduced in Release 3.5.0. See the ?$paranum>Configuring Directed Nonlimiting CAC Policies? section on page 7-37 for information about this command.

The following example shows that the SBC is configured to convert an AS bandwidth line format into a TIAS bandwidth line format on the offerer-side adjacency (caller side), and to convert a TIAS bandwidth line format into an AS bandwidth line format on the answerer-side adjacency (callee side):

```
Router# configure terminal
Router(config)# sbc mysbc
Router(config-sbc)# sbe
Router(config-sbc-sbe)# cac-policy-set 1
Router(config-sbc-sbe-cacpolicy)# cac-table 1
Router(config-sbc-sbe-cacpolicy-cactable)# entry 1
Router(config-sbc-sbe-cacpolicy-cactable-entry)# caller-bandwidth-field as-to-tias
Router(config-sbc-sbe-cacpolicy-cactable-entry)# callee-bandwidth-field tias-to-as
```

The following example lists detailed information, including caller and callee bandwidth field information, for entry 1 of CAC table 1:

```
Router# show sbc mysbc sbe cac-policy-set 1 table 1 entry 1

SBC Service "mysbc"
```

```
CAC Policy Set 1
  Active policy set: Yes
  Description:
  Averaging period: 60 sec
  First CAC table: 1
  First CAC scope: global

  Table name: cacTable
    Description:
    Table type: policy-set                      Total call failures: 0

    Entry 1
    CAC scope:
    Action: CAC complete                    Number of calls rejected: 0
    Max calls per scope:    Unlimited       Max call rate per scope: Unlimited
    Max in-call rate:       Unlimited       Max out-call rate:       Unlimited
    Max reg. per scope:     Unlimited       Max reg. rate per scope: Unlimited
    Max channels per scope: Unlimited       Max updates per scope:   Unlimited
    Early media:            Allowed         Early media direction:   Both
    Early media timeout:    None            Transcoder per scope:    Allowed
    Callee Bandwidth-Field: TIAS-to-AS      Caller Bandwidth-Field:  AS-to-TIAS
    Media bypass:                    Allowed
    Media flag:                      Ignore bandwidth-fields (b=)
    Renegotiate Strategy:            Delta
    Max bandwidth per scope:         Unlimited
    SRTP Transport:                  Trusted-Only (by default)
    Caller hold setting:             Standard
    Callee hold setting:             Standard
    Caller privacy setting:          Never hide
    Callee privacy setting:          Never hide
    Caller voice QoS profile:        Default
    Caller video QoS profile:        Default
    Caller sig QoS profile:          Default
    Callee voice QoS profile:        Default
    Callee video QoS profile:        Default
    Callee sig QoS profile:          Default
    Restrict codecs to list:         Default
    Restrict caller codecs to list:  Default
    Restrict callee codecs to list:  Default
    Caller inbound SDP policy:       None
    Caller outbound SDP policy:      None
    Callee inbound SDP policy:       None
    Callee outbound SDP policy:      None
```

Per-Adjacency Codec String Interworking

The following sections are in the Per-Adjacency Codec String Interworking feature:

- Information about Per-Adjacency Codec String Interworking, page 18-24
- Restrictions for Per-Adjacency Codec String Interworking, page 18-24
- Configuring Per-Adjacency Codec String Interworking, page 18-24
- Configuration Example for Per-Adjacency Codec String Interworking, page 18-30

Information about Per-Adjacency Codec String Interworking

From Cisco IOS Release 3.2S, the SBC can interpret non-standard SDP, and convert codecs between different non-standard forms of SDP or convert non-standard SDP to standard SDP, so that different non-standard devices can interwork through the SBC.

The SBC works through each codec in the message and determines whether the codec name received on the corresponding inbound SDP is a standard name or a variant, and then converts the codec:

- If the codec is standard, SBC searches through the variant list looking for a matching variant of that standard codec. The matching variant, if found, is converted or passed through unchanged.
- If the codec is a variant, SBC searches through the profile to see if that variant is listed.
  - Listed variant passes through unchanged
  - Unlisted variant can get converted to a matching variant of the same standard codec.
  - Unlisted variant that does not have a matching variant of the same standard codec gets converted to the standard representation

In the following scenarios, the codec convert should be avoided and SBC should use the standard SDP form:

- To ensure consistency of the H.248 interface, SDP fragments sent by SBC-SIG in H.248 commands to the media gateways (MGs) must use standard representations. The H.248 interface must be specified to facilitate interoperability with MGs and make it easy for the third party MGs to implement the SBC H.248 profile.
- To ensure consistency of the billing interface, the SDP fragments stored by SBC-SIG in the XML billing records must use standard representations. It is easy for the third party billing servers to parse XML billing records.
- Codec strings do not appear in H.245 signaling messages, an enumerated type is used to represent the codec. Therefore, the codec convert is only applicable for SIP outbound adjacency.

Restrictions for Per-Adjacency Codec String Interworking

The Per-Adjacency Codec String Interworking feature has the following restrictions:

- When a particular variant is used for a given codec on passing through an Offer, the same variant may not be used when passing through the Answer.
- For a given side of the call, SBC cannot be configured to interpret the incoming SDP using one variant but convert outgoing SDP based on another.
- You cannot define a variant that uses a standard IANA codec string because the SBC supports only those variants that use non-standard strings.
- If two variants map to the same standard codec, the transcoder does not convert between them. For example, the SBC cannot transcode between G7231H and G7231L, even though the endpoints can perceive those to be different codecs.

Configuring Per-Adjacency Codec String Interworking

This section explains the following configurations for Per-Adjacency Codec String Interworking feature:

- Configuring Codec Variant Conversion, page 18-25

- Configuring Codec on CAC Policy Set, page 18-27

Configuring Codec Variant Conversion

This task explains how to configure codec variant conversion on the SBC:

SUMMARY STEPS

1. **configure terminal**
2. **sbc** *service-name*
3. **sbe**
4. **codec system** *sys-codec* **id** *payload-id*
5. **fmtp** *fmtp-string*
6. **exit**
7. **codec variant codec** *variant-name*
8. **variant** *variant-codec-encoded-name*
9. **standard** *standard-codec-name*
10. **fmtp** *fmtp-string*
11. **exit**
12. **codec variant profile** *profile-name*
13. **variant** *variant-name*
14. **end**
15. **show sbc** *service-name* **sbe codecs variant** [**profile**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Router# configure`	Enables global configuration mode.
Step 2	**sbc** *service-name* **Example:** `Router(config)# sbc mysbc`	Enters the SBC service mode. • Use the *service-name* argument to define the name of the service.
Step 3	**sbe** **Example:** `Router(config-sbc)# sbe`	Enters the SBE entity mode within an SBC service.

	Command or Action	Purpose
Step 4	**codec system** *sys-codec* **id** *payload-id* **Example:** `Router(config-sbc-sbe)# codec system G723 id 4`	Enters the codec definition mode to modify an existing codec. • *sys-codec*—The codec included in the SBC. • **id** *payload-id*—Static payload id. Value can be from 0 to 96.
Step 5	**fmtp** *fmtp-string* **Example:** `Router(config-sbc-sbe-codec-def)# fmtp annexa=yes`	Configures the default value of Format-Specific Parameters (FMTP) for SDP. • *fmtp-string*—The FMTP string for SDP, in the name=value format. **Note** To view the default FMTP values associated with variants, use the **show sbc sbe codecs variant** command.
Step 6	**exit** **Example:** `Router(config-sbc-sbe-codec-def)# exit`	Exits from the codec definition mode and enters into the SBE entity mode.
Step 7	**codec variant codec** *variant-name* **Example:** `Router(config-sbc-sbe)# codec variant codec G723-H-1`	Enters the codec variant mode to configure, modify, or delete a codec variant. • *variant-name*—The codec variant name.
Step 8	**variant** *variant-codec-encoded-name* **Example:** `Router(config-sbc-sbe-codec-var-codec)# variant G723-H-1`	Defines the encoded codec variant name. • *variant-codec-encoded-name*—The variant nonstandard codec string. **Note** '#' is reserved for base variants. Therefore, the variant name cannot start with '#'
Step 9	**standard** *standard-codec-name* **Example:** `Router(config-sbc-sbe-codec-var-codec)# standard G723`	Defines the standard codec variant name. • *standard-codec-name*—The standard system codec name.
Step 10	**fmtp** *fmtp-string* **Example:** `Router(config-sbc-sbe-codec-var-codec)# fmtp bitrate=6.3`	Define the FMTP parameters for the codec variant. • *fmtp-string*—The FMTP string in the name=value format. **Note** To view the default FMTP values associated with variants, use the **show sbc sbe codecs variant** command.
Step 11	**exit** **Example:** `Router(config-sbc-sbe-codec-var-codec)# exit`	Exits the codec variant mode and enters into the SBE entity mode.

	Command or Action	Purpose

Step 12	**`codec variant profile`** *`profile-name`*
	Example:
	`Router(config-sbc-sbe)# codec variant profile Profile-1`
		Enters the codec variant profile mode to configure a codec variant profile.
		- *profile-name*—The codec profile name.

Step 13	**`variant`** *`variant-name`*
	Example:
	`Router(config-sbc-sbe-codec-var-prof)# variant G723-H-1`
		Defines the codec variant name.
		- *variant-name*—The codec variant name.
		Note '#' is reserved for base variants. Therefore, the variant name cannot start with '#'

Step 14	**`end`**
	Example:
	`Router(config-sbc-sbe-codec-var-prof)# end`
		Exits the codec variant profile mode and enters in the privileged EXEC mode.

Step 15	**`show sbc`** *`service-name`* **`sbe codecs variant`** `[`**`profile`**`]`
	Example:
	`Router# show sbc mySBC sbe codecs variant`
		Displays information about codec variants on the SBC.
		- **profiles**—Displays information about codec variant profiles.

Configuring Codec on CAC Policy Set

This task shows how to enable codec convert and configure codec variant profile on a CAC policy set. When the codec variant conversion is enabled or disabled, the following events occur:

- If the codec variant conversion is disabled, the SBC does not take into account the specified variant profile. All the codecs that have been passed are left in their original representation, and any new codecs added by the SBC are added with their standard representation.
- If the codec variant conversion is enabled but the variant profile is not configured, all the codecs are converted to its standard representation.
- If the codec variant is enabled and the variant profile is configured, any codecs matched by the profile are converted to the appropriate variant representation, and any codecs that is not matched by the variant profile is converted to its standard representation.

Note The **caller** and **callee** commands have been used in this procedure. In some scenarios, the **branch** command can be used as an alternative to the **caller** and **callee** command pair. The **branch** command has been introduced in Release 3.5.0. See the ?$paranum>Configuring Directed Nonlimiting CAC Policies? section on page 7-37 for information about this command.

SUMMARY STEPS

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

6. **table-type {policy-set | limit {***list of limit tables}}*
7. **entry** *entry-id*
8. **caller codec convert**
9. **callee codec convert**
10. **caller codec profile** *profile-name*
11. **callee codec profile** *profile-name*
12. **exit**
13. **exit**
14. **complete**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`configure terminal`**
	Example:
	`Router# configure`
		Enables global configuration mode.

Step 2	**`sbc`** *`service-name`*
	Example:
	`Router(config)# sbc mysbc`
		Enters the SBC service mode.
		Use the *service-name* argument to define the name of the service.

Step 3	**`sbe`**
	Example:
	`Router(config-sbc)# sbe`
		Enters the SBE entity mode within an SBC service.

Step 4	**`cac-policy-set`** *`policy-set-id`*
	Example:
	`Router(config-sbc-sbe)# cac-policy-set 1`
		Enters the CAC policy set configuration mode within an SBE entity, creating a new policy set if necessary.

Step 5	**`cac-table`** *`table-name`*
	Example:
	`Router(config-sbc-sbe-cacpolicy)# cac-table StandardListByAccount`
		Enters the CAC table mode for configuration of an admission control table (creating one, if necessary) within the context of an SBE policy set.

Step 6	**`table-type {policy-set	limit`** `{`*`list of limit tables`*`}}` **Example:** `Router(config-sbc-sbe-cacpolicy-cactable)# table-type policy-set`	Configures the table type of a CAC table within the context of an SBC policy set.

	Command or Action	Purpose
Step 7	**entry** *entry-id* **Example:** `Router(config-sbc-sbe-cacpolicy-cactable)# entry 1`	Enters the mode to create or modify an entry in an admission control table.
Step 8	**caller codec convert** **Example:** `Router(config-sbc-sbe-cacpolicy-cactable-entry) # caller codec convert`	To enable or disable codec variant conversion at the caller side.
Step 9	**callee codec convert** **Example:** `Router(config-sbc-sbe-cacpolicy-cactable-entry) # callee codec convert`	To enable or disable codec variant conversion at the callee side.
Step 10	**caller codec profile** *profile-name* **Example:** `Router(config-sbc-sbe-cacpolicy-cactable-entry) # caller codec profile`	To specify a codec variant profile at the caller side. • *profile-name*—The codec variant profile name.
Step 11	**callee codec profile** *profile-name* **Example:** `Router(config-sbc-sbe-cacpolicy-cactable-entry) # callee codec profile`	To specify a codec variant profile at the callee side. • *profile-name*—The codec variant profile name.
Step 12	**exit** **Example:** `Router(config-sbc-sbe-cacpolicy-cactable-entry) # exit`	Exits from the **entry** mode and enters into the **cactable** mode.
Step 13	**exit** **Example:** `Router(config-sbc-sbe-cacpolicy-cactable)# exit`	Exits from the **cactable** mode and enters into the **cacpolicy** mode.
Step 14	**complete** **Example:** `Router(config-sbc-sbe-cacpolicy)# complete`	Completes the CAC policy set when you have committed the full set.

Configuration Example for Per-Adjacency Codec String Interworking

The following example shows how to configure the Per-Adjacency Codec String Interworking feature when caller supports G723 bitrate 6.3 annexa=no codec and callee supports G.723.1 codec variant:

```
configure terminal
sbc MySBC
sbe
codec variant codec PCMU.1
  standard PCMU
  variant PCMU.1
  exit
codec variant profile pcmu-var
  variant PCMU.1
  variant #G.723.1/H
  exit
  cac-policy-set 2
    first-cac-table codec-convert
    first-cac-scope src-adjacency
    cac-table codec-convert
     table-type limit src-adjacency
     entry 1
      match-value CallMgrA
      callee codec profile pcmu-var
      callee codec convert
      media police strip
      action cac-complete
    complete
    end
```