FRAFOS ABC SBC Reference Book

Release 5.6

FRAFOS GmbH

May 06, 2026

Table of Contents

Chapter 1

Reference of Actions

The actions are grouped as follows:

SIP Mediation

Request URI manipulation

- Set RURI
- Prefix RURI user
- Set RURI user
- Append to RURI user
- Strip RURI User
- Set RURI Host
- Set RURI Parameter

To/From manipulation

- Set From
- Set From display name
- Set From User
- Set From Host
- Set To
- Set To Display Name
- Set To User
- Set To Host

Contact HF manipulation

- Set Contact-URI user
- Set Contact-URI host
- Set Contact-HF parameter whitelist/blacklist
- Set Contact-URI parameter whitelist/blacklist
- Forward Contact-HF parameters
- Forward Contact-URI parameters
- Keep Contact user
- Add Dialog Contact Parameter
- Add Contact header field parameter

Authorization

- UAC auth
- UAS auth

Common header manipulation

- Remove Header
- Add Header
- Replace header value
- Replace header value (on leg)
- Insert or Replace header (on leg)
- Set header whitelist
- Set header blacklist
- Update Supported header
- Update Require header
- Update Allow header
- Replace URI header user
- Replace URI header host
- Replace headers of URI header
- Insert or replace headers of URI header
- Diversion to History-Info
- Set Max Forwards
- Map Replaces header
- Forward Via-HFs
- Add X-Org-ConnID header

Session timers

- Enable SIP Session Timers (SST) - caller leg
- Enable SIP Session Timers (SST) - callee leg

Others

- Absorb Re-INVITEs (on leg)
- Absorb UPDATEs (on leg)
- Relay 503 Reply (on leg)
- Reply In-Dialog Request (on leg)
- Translate Reply Code
- Set SIP Timers
- Handle INVITE with Replaces header
- Pin TLS Certificate To Dialog (on leg)
- Set Content Type whitelist/blacklist
- Insert or Replace SIP Message Body (on leg)

- Enable transparent dialog IDs
- Call transfer handling
- Replace SIP Message Body (on leg)

SDP Mediation

- Set CODEC Whitelist
- Set CODEC Blacklist
- Set CODEC Preferences
- Set SDP attribute whitelist
- Set SDP attribute blacklist
- Set SDP bandwidth limit
- Set Media whitelist
- Set Media blacklist
- Drop early media
- Drop SDP from 1xx replies
- Insert or Replace SDP Session Attribute (on leg)
- Replace SDP Session Attribute (on leg)
- Insert or Replace SDP Media Attribute (on leg)
- Replace SDP Media Attribute (on leg)
- Disable SDP Media
- Remove SDP Media Attribute (on leg)
- Insert or Replace SDP Payload Attribute (on leg)
- Replace SDP Payload Attribute (on leg)
- Limit telephony event list (on leg)
- DTLS Setup Preference (on leg)

Monitoring and Logging

- Increment custom counter
- Log received traffic
- Log Event
- Set log level
- Log Message
- Log Message for Replies
- Log to grey list
- Disable privacy monitor mode

Traffic Shaping

- Limit parallel calls
- Limit CAPS
- Limit Bandwidth per Call
- Limit Bandwidth
- Set call Timer

Media Processing

- Enable RTP anchoring
- Restrict media IP to signaling IP (on leg)
- Force RTP/SRTP
- SRTP Fallback to RTP (on leg)
- Activate audio recording
- Activate transcoding
- Process RTP Header Extension
- Join meet-me conference
- Meet-me conference set PIN
- Refuse call with audio prompt
- Play prompt on final response
- Generate Ring-Back Tone
- Activate Music On Hold

DTMF handling

- Convert DTMF to AVT RTP
- Convert DTMF to SIP INFO
- Activate Inband DTMF Detection
- DTMF Termination Same SSRC (on leg)
- DTMF Termination Stable Duration Increments (on leg)

SIP Dropping

- Reply to request with reason and code
- Drop request
- Allow unsolicited NOTIFYs

Scripting

- Set Call Variable

Register Processing

- Enable REGISTER caching
- Retarget R-URI from cache
- REGISTER throttling
- Save REGISTER contact in registrar
- Restore contract from registrar

External Interaction

- ENUM query
- Read call variables over REST
- Read call variables from table

NAT Handling

- Enable dialog NAT handling

Other

- Support serial forking proxy
- Fork

1.1 SIP Mediation

1.1.1 Set RURI

Set request URI of the outgoing request to a new value.

Can be used in A and C rules.

> **Info**
>
> This action affects outgoing, dialog initiating request only. In-dialog requests in both directions follow SIP protocol and use content of remote peer's Contact header for building the request URI.

> **Warning**
>
> Using an invalid value will lead to processing error and outbound request wouldn't be sent. "Parser failed on generated request" error will be logged in SEMS log in such case.

Parameters

new URI

New value of request URI.

Accepts replacement expressions.

1.1.2 Prefix RURI user

Prefix user part of request URI.

The values are cumulated thus using this action twice will lead to adding two prefixes.

Adding a prefix (for example `AA`) to an URI without username part (`sip:domain.com`) will create the user part (`sip:AA@domain.com`).

Can be used in A and C rules.

Parameters

prefix string

Prefix that should be prepended to the user part of request URI.

Accepts replacement expressions.

Info

This action affects outgoing, dialog initiating request only. In-dialog requests in both directions follow SIP protocol and use content of remote peer's Contact header for building the request URI.

Warning

Using value that will break R-URI syntax will lead to processing error and outbound request wouldn't be sent. "Parser failed on generated request" error will be logged in SEMS log in such case.

1.1.3 Set RURI user

Replace user part of request URI.

Parameters

new user part

1.1.4 Append to RURI user

Add a suffix to user part of request URI. The result is accumulated if actions is used multiple times.

Parameters

suffix

1.1.5 Strip RURI User

Remove leading characters of user part of request URI.

Parameters

number of leading characters

1.1.6 Set RURI Host

Replace hostport part of request URI.

Parameters

new host part

1.1.7 Set RURI Parameter

Set request URI parameter.

Parameters

parameter name

parameter value

1.1.8 Set From

Replace From Header Field Value.

Parameters

From HF value

1.1.9 Set From display name

Replace From Display name.

Parameters

new From Display name

1.1.10 Set From User

Replace user part of From URI.

Parameters

new From user part

1.1.11 Set From Host

Replace hostport part of From URI.

Parameters

new From hostname

1.1.12 Set To

Replace To Header Field Value.

Parameters

To HF value

1.1.13 Set To Display Name

Replace To Display name.

Parameters

new To Display name

1.1.14 Set To User

Replace user part of To URI.

Parameters

new To user part

1.1.15 Set To Host

Replace hostport part of To URI.

Parameters

new To hostname

1.1.16 Set Contact-URI user

Set the Contact-HF URI user part used for the dialog.

Available since: 4.2

1.1.17 Set Contact-URI host

Override host part of Contact URI used by ABC SBC in appropriate direction.

If this action is not used, ABC SBC uses IP address of appropriate signaling interface (or "Public IP address" if configured) to compose its Contact header. With this action, the host part of generated Contact URI is overridden with the configured value.

Can be used in A and C rules. If used in A rules, it overrides SBC's Contact header in requests or replies (even in-dialog ones) being sent towards caller. If used in C rules, it overrides SBC's Contact header in messages sent towards callee.

Available since: 4.5

> ⚠ **Warning**
>
> The Contact header field is used by peers to send in-dialog messages to the ABC SBC. If the syntax is broken or if it doesn't point to the appropriate signaling interface, in-dialog messages couldn't be sent by peers (i.e. for example BYE won't be properly delivered and thus calls couldn't be properly terminated).

Parameters

Host

New value of Contact header host.

Replacement expressions and back-references are allowed.

Apply on

Can be used to control on which message type (request, reply or both) the modification is to be applied to.

Available since 5.1.

Only on reply codes

Can be used to control on which reply codes the modification is to be applied to.

If empty, replies with code less than 300 (i.e. provisional and success class responses) are affected.

This is only effective if 'Apply on' is set to a value that will affect replies.

Available since 5.1.

1.1.18 UAC auth

Authenticate on behalf of UAC against an UAS. Any request passing this action and challenged to authenticate by a downstream server will be resent with credentials passed in the action's parameters.

> **Info**
>
> Note that the input fields support replacement expressions. If i.e. password contains special characters such as `$`, they need to be escaped with a backslash.

Parameters

username

password

realm

1.1.19 UAS auth

Authenticate a UAC against the SBC. Either HA1 or password can be provisioned on the SBC; HA1 is safer as the plaintext password does not need to be saved on the SBC. The HA1 can be calculated as MD5(username:realm:password) on the command line. Can be used together with provisioned tables and the Save REGISTER contact in registrar action to create a full registrar.

Parameters

username

realm

H(A1) or password

1.1.20 Remove Header

Removes all occurrences of a header field. The action is applied to initial message, newly added header fields are not removed.

Parameters

header field name

1.1.21 Add Header

Add a new Header Field to a request.

> **Info**
>
> '100 Trying' replies are generated by the SBC. So an action on C-rules with `direction = A leg` will not work on 100-replies because they are not coming from the B-leg. Action on A-rules will work as fine with respect to 100-replies.

> **Info**
>
> Replacement expressions are evaluated once at the beginning of the call (initial request) and the result is re-used throughout the call.

Parameters

HF Name

HF Value

Request or reply

Can be used to control on which type of messages the header will be added on.

Available since 5.1.

Initial or in-dialog

Can be used to choose to only add the header on initial or in-dialog requests, or both.

Available since 5.1.

Direction

Can be used to choose whether to add the header on messages going towards a-leg, b-leg or both.

Available since 5.1.

1.1.22 Replace header value

Replaces matching header field values based on regular expression search and replace.

Parameters

header name

search

replace with

Replacement expressions are allowed, so for example a call variable value may be used here (for example: `$V(gui.fullname)`).

Also, with "replace with", one can use regular expression back-references to use parts of the expression in "match" parameter.

I.e. to replace host part in a header containing a URI, search for `^<sip:([^@]*)@[^?;]*(.*)>` and replace with `<sip:\$1@a.b.c.d\$2>` can be used.

Note that you can only back-reference from 1 to 9 sub-matches, meaning that `\$123` will replace as `<sub-match-1>23`.

1.1.23 Replace header value (on leg)

Same as Replace header value but acts on messages on call leg only.

E.g. putting a rule on A rules of CA1:

```
[CA1] INVITE -> [SBC] -> [CA2] 200 OK -> [SBC] rule-applied -> [CA1]
```

E.g. putting a rule on C rules of CA2:

```
[CA1] INVITE -> [SBC] rule-applied -> [CA2] 200 OK -> [SBC] -> [CA1]
```

Available since: 4.6

Parameters

header name

search

replace with

1.1.24 Insert or Replace header (on leg)

Tries to insert a header field to messages. Unless "replace existing" is enabled, a new header will be added even if a header with the same name exists. If "replace existing" is enabled, the header is replaced with the given value.

Available since: 4.6

Parameters

header name

header value

Replacement expressions and regular expression back-references are allowed.

replace existing

1.1.25 Absorb Re-INVITEs (on leg)

Absorb re-INVITEs coming from the leg if they are considered identical to the previous (re-)INVITE. The decision is done based on:

- All headers match except the following ignored headers: Call-Id, Contact, Content-Length, Content-Type, From, To, Via, RAck, CSeq, Route and Record-Route.
- If the request has a body, the body type is SDP, and the SDP is considered identical (see below).

When SDP is being checked, the SDP of the session is considered. I.e. SDP negotiated via late-oa, or an UPDATE affects this.

For comparing body, if there's application/sdp, only the SDP is taken into account. Within the SDP, only s=, c=, other session-level a=, m= and everything media-level is taken into account. If there is no body on an incoming request, then its body is considered equal to the previous one. If there is a body but the body does not contain application/sdp, then it is considered not-equal to the previous one.

Available since: 4.6.

Parameters

Session-Expires Percentile

If Session-Expires Percentile is set, the INVITE will not be absorbed if the time elapsed has exceeded the set value since the last relayed INVITE. I.e. if percentile is set to 10 and last (re-)INVITE has Session-Expires: 90, then a re-INVITE will be relayed if more than 9 seconds has passed since the last relayed (re-)INVITE even if it is considered identical.

If Session-Expires percentile is empty, then the SBC will absorb re-INVITEs even if they were supposed to refresh the session.

If there is no Session-Expires header in a received UAC Request, then Session-Expires percentile is not checked. This ends up with the same effect as not setting the Session-Expires percentile.

If there was no Session-Expires header in the last received UAC Request and it was not absorbed, then the SBC will not check for the validity of Session-Expires percentile on the following requests. This ends up with the same effect as not setting the Session-Expires percentile.

If Absorb UPDATEs action is also used, then Session-Expires calculations are done in a common way on both INVITEs and UPDATEs.

Ignore Headers

If Ignore Headers is set, then request headers do not affect the decision on absorbing the INVITE or not. This effectively means that only the SDP is compared to previously sent SDPs for equality. Note that the Session-Expires parameter is still honoured if set.

Ignore Body

If Ignore Body is set, then request body does not affect the decision on absorbing the INVITE or not.

1.1.26 Absorb UPDATEs (on leg)

Absorb UPDATEs coming from the leg if they are considered identical to the previous UPDATEs. The decision is done based on:

Parameters and behavior is the same as Absorb Re-INVITEs (on leg).

Note that the first UPDATE will never be absorbed, unless Ignore Headers parameter is enabled. Headers of the UPDATE request are compared separately and the first UPDATE will mark the initial state for the previous headers.

Available since: 5.4.

1.1.27 Relay 503 Reply (on leg)

Normally, per **RFC 3261 Section 16.7**, 503 replies are converted to 500 before sending the reply out to the CA. With this action, 503 replies are relayed to the call leg it is on.

Available since: 5.1.

1.1.28 Reply In-Dialog Request (on leg)

Reply In-Dialog requests matching "Method" (case-insensitive) with a reply with the code "Code".

Parameters

Method

Code

1.1.29 Set header whitelist

Removes all but mandatory and white-listed header-fields.

The list is applied to the final appearance of the INVITE request after all A and C rules have been processed.

Parameters

header-field names

Comma-separated, case-insensitive list of header field names.

> ⚠ **Warning**
>
> compact form needs to be mentioned explicitly!

1.1.30 Set header blacklist

Removes all blacklisted header-fields.

The list is applied to the final appearance of the INVITE request after all A and C rules have been processed.

Parameters

header-field names

Comma-separated, case-insensitive list of header field names.

> ⚠ **Warning**
>
> compact form needs to be mentioned explicitly!

1.1.31 Insert or Replace SIP Message Body (on leg)

Allows inserting or modifying SIP message body based on mime type.

Available since: 5.4

Parameters

Mime-type

Mime type to match. Replacement expressions and back-references are supported. The mime-type `application/sdp` cannot be used here and the action will not be applied if a replacement results in that.

Pattern

RegExp pattern to match. If `Replace with` is enabled, matched part will be replaced with `Value`. If `Replace with` is not enabled, this is ignored. Replacement expressions and back-references are supported.

Value

When `Replace with` is enabled and given mime-type exists, matched part is replaced with the given value. When `Replace with` is enabled and given mime-type does not exist, `Pattern` is ignored and sets the content-type (or makes the message multipart and inserts a new part if the message already has a body) and content to the given value. When `Replace with` is not enabled, `Pattern` is ignored and sets the content-type (or makes the message multipart and inserts a new part if the message already has a body) and content to the given value. Replacement expressions and back-references are supported.

Replace with

When checked, if given mime type already exists, runs a replacement on it instead of inserting. Replacement expressions and back-references are supported.

1.1.32 Replace SIP Message Body (on leg)

Allows modifying SIP message body based on mime type.

Available since: 5.4

Parameters

Mime-type

Mime type to match. Replacement expressions and back-references are supported. The mime-type `application/sdp` cannot be used here and the action will not be applied if a replacement results in that.

Pattern

RegExp pattern to match. Replacement expressions and back-references are supported.

Value

Value to replace the matched part with. Replacement expressions and back-references are supported.

1.1.33 Update Supported header

Allows simplified manipulation with Supported header field content.

Available since: 4.5.

Parameters

operator

Specifies how to use given list of tags.

Add tags

Add the listed tags to the current list of supported tags.

Remove tags

Remove listed tags from the current list of supported tags.

Set tags

Overwrite current list of supported tags with the listed ones.

Whitelist tags

Remove tags that are not listed. Available since: 5.5

comma-separated list of option tags

1.1.34 Update Require header

Allows simplified manipulation with Require header field content.

Available since: 4.5

Parameters

operator

Specifies how to use given list of tags.

Add tags

Add the listed tags to the current list of required tags.

Remove tags

Remove listed tags from the current list of required tags.

Set tags

Overwrite current list of required tags with the listed ones.

comma-separated list of option tags

1.1.35 Update Allow header

Allows simplified manipulation with Allow header field content.

Info

"Add" operator will not add unless Allow header already exists, set via "Set" operator or "Default tags" are specified.

Available since: 4.6.

Parameters

operator (Add / Remove / Set tags)

comma-separated list of option tags

Direction

Apply on

Default tags

1.1.36 Replace URI header user

Allows modifying "user" part on headers containing an URI. I.e. `Refer-to: sip:USER@host`.

Available since: 5.0.

Parameters

Header name

Search

Replace with

1.1.37 Replace URI header host

Allows modifying "host:port" part on headers containing an URI. I.e. `Refer-to: sip:user@HOST:PORT`.

Available since: 5.0.

Parameters

Header name

Search

Replace with

1.1.38 Replace headers of URI header

Allows modifying headers in headers containing URIs.

I.e. Call-ID in `Refer-to: <sip:user@host?Call-ID=55432%40alicepc.atlanta.example.com>` can be manipulated with "header name = refer-to", "name of the header in URI = call-id", "Search = 432@alice", "replace with = 433@bob".

Available since: 5.0.

Parameters

Header name

Name of the header in URI

Search

Replace with

1.1.39 Insert or replace headers of URI header

Allows modifying headers in URI of headers containing a URI.

I.e. NEW-hdr in `Refer-to: <sip:user@host?Call-ID=55432%40alicepc.atlanta.example.com&NEW-hdr=value>` can be added with this.

Parameters

Header to modify

Header name

Header value

Replace if exists

1.1.40 Add Dialog Contact Parameter

Add parameters to the Contact URI generated by the SBC. Acts on all out-of-dialog, dialog-initiating or in-dialog messages.

I.e. `new-param` in `Contact: <sip:user@host;new-param=value>` can be added with this.

Parameters

Leg: A or B parameter name parameter value

1.1.41 Add Contact header field parameter

Add Contact header field parameters to the Contact header generated by the SBC. Acts on all out-of-dialog, dialog-initiating or in-dialog messages.

The given value parameter needs to be already escaped.

Note that this action will not work together with REGISTER messages when `Save REGISTER contact in registrar` or `Enable REGISTER caching` actions are used.

Note that this action will not work on provisional replies when it is in C-rules and the `Leg` parameter is `A`.

I.e. `new-param` in `Contact: <sip:user@host>;new-param=value` can be added with this.

Parameters

Leg: A or B parameter name parameter value

1.1.42 Set Contact-HF parameter whitelist/blacklist

Specify which Contact header field parameters in incoming request to forward downstream.

Parameters

comma-separated list of parameter names

1.1.43 Set Contact-URI parameter whitelist/blacklist

Specify which Contact URI parameters in incoming request to forward downstream.

Available since: 4.6.

Parameters

comma-separated list of parameter names

1.1.44 Forward Contact-HF parameters

Forward all Contact header field parameters "as is" downstream.

1.1.45 Forward Contact-URI parameters

Forward all Contact URI parameters "as is" downstream.

Available since: 4.6.

1.1.46 Keep Contact user

Keep Contact URI user part as received from the other peer in Contact header generated by ABC SBC.

Without this action, ABC SBC generates its Contact URI with username part representing the dialog identifier. If this action is used, the username part from incoming Contact URI is preserved and used in SBC's Contact URI towards the other peer and new Contact URI parameter `dlg-id` is added and used to identify the dialog instead of the URI username.

Can be used in A and C rules and affects the appropriate call leg only.

If this action is used in A rules, the callee's username in Contact URI is preserved and sent in Contact header in messages towards caller. For example:

Caller sends INVITE with its Contact header:

```
INVITE sip:104@vku-test.com SIP/2.0
...
Contact: <sip:101@192.168.13.221:6010;ob>
...
```

ABC SBC forwards the INVITE with usual Contact header ("Keep Contact user" is not used in C rules):

```
INVITE sip:104@192.168.13.221:6040;ob SIP/2.0
...
Contact: <sip:21F67A8F-64DF20AB0005698E-923FF6C0@192.168.13.51;transport=udp>
...
```

Callee replies with its Contact:

```
SIP/2.0 200 OK
...
Contact: <sip:104@192.168.13.221:6040;ob>
...
```

ABC SBC forwards the Contact URI username to caller ("Keep Contact user" is used in A rules) and adds `dlg-id` parameter:

```
SIP/2.0 200 OK
...
Contact: <sip:104@192.168.13.51;dlg-id=4D1B3203-64DF20AB00055FCD-B833D6C0;transport=tcp>
...
```

If it is used in C rules, the caller's username is used in Contact header in messages towards callee. For example:

Caller sends INVITE with its Contact header:

```
INVITE sip:104@vku-test.com SIP/2.0
...
Contact: <sip:101@192.168.13.221:6010;ob>
...
```

ABC SBC forwards the Contact URI username to callee ("Keep Contact user" is used in C rules) and adds `dlg-id` parameter:

```
INVITE sip:104@192.168.13.221:6040;ob SIP/2.0
...
Contact: <sip:101@192.168.13.51;dlg-id=386CF1E5-64DF2A70000DDF70-921FD6C0;transport=udp>
...
```

Callee replies with its Contact:

```
SIP/2.0 200 OK
...
Contact: <sip:104@192.168.13.221:6040;ob>
...
```

ABC SBC forwards the reply with usual Contact header ("Keep Contact user" is not used in A rules):

```
SIP/2.0 200 OK
...
Contact: <sip:01E3A1EE-64DF2A70000DD992-B833D6C0@192.168.13.51;transport=tcp>
...
```

1.1.47 Translate Reply Code

Translate SIP reply codes to other value.

Parameters

matching reply code

new reply code

new reason phrase

1.1.48 Set Max Forwards

Reset the number of hops a request can be forwarded to specified value.

Parameters

the new value of Max-Forwards header field

1.1.49 Enable transparent dialog IDs

Enforce use of the same dialog IDs on both sides of a call.

Parameters

To-tag

Controls To-tag handling. Can have following values:

Stick to first received to-tag

Keeps the first seen to-tag in the early responses throughout the rest of the dialog, even if it changes in the final reply.

Re-set to-tag with final reply

Will switch the to-tag from early to established dialog (on first final reply sent to caller).

1.1.50 Forward Via-HFs

Force the SBC to keep the Via header fields while forwarding the request.

1.1.51 Diversion to History-Info

Converts SIP Diversion header-field into History-Info.

1.1.52 Call transfer handling

Defines the mode in which REFERs are handled: rejection, local processing or forwarding.

Parameters

Mode

REFER processing mode. Can be one of

REFER pass-through

Handle REFER internally

Reject REFER

Reconnect on all failures during unattended transfer

Reconnect if transfer ends in 4xx during unattended transfer.

Do not terminate after unattended transfer

Do not terminate referrer leg when the unattended transfer completes.

Only NOTIFY 100 & final sip replies

Disables relaying of provisional replies of transferee to referrer as NOTIFY messages. It can come useful in scenarios where backup CA agent is tried and provisional replies of latter CA might confuse the referrer.

1.1.53 Set SIP Timers

Allows setting SIP timers per call.

Parameters

SIP Timers

Failover reduce factor

This parameter is used to divide B, F & M timers when destination call agent has a backup CA. This allows for a faster failover. Leaving it empty uses the default value of 4.

1.1.54 Handle INVITE with Replaces header

Activates internal processing of INVITE with Replaces header.

1.1.55 Map Replaces header

Activates mapping of dialog identifiers in INVITE with Replaces.

1.1.56 Pin TLS Certificate To Dialog (on leg)

This action causes remembering the initial client certificate that's used while initiating the dialog and rejects any in-dialog request that do not use the same certificate.

This action requires "Verify peer certificate" to be enabled on the TLS Profile of the signaling interface.

Note that non-TLS messages, messages with no associated TLS client certificates or messages with different different certificates compared to the pinned one will be:

- Rejected with 403 if it is an initial request.
- Rejected with 481 if it is an in-dialog request.
- Dropped if it is a reply or an ACK.

When used in A rules:

- If SHA256 fingerprint is empty, then the fingerprint of the certificate used in the initial request is pinned.
- If SHA256 fingerprint is given, then it is pinned for the dialog and the certificate used in the initial request will also be compared against it.

When used in C rules, SHA256 fingerprint must be given.

In order to get the SHA256 fingerprint of a certificate, the following command may be used: `openssl x509 -noout -fingerprint -sha256 -inform pem -in <CERT>`

Available since: 5.2.

Parameters

SHA256 fingerprint

1.1.57 Set Content Type whitelist/blacklist

Specifies which SIP payload types (such as SDP) will be permitted.

Parameters

comma-separated list of content types

1.1.58 Enable SIP Session Timers (SST) - caller leg

Enforce the use of session timers for the caller. Support for session timers is not advertised to the callee (the `timer` extension is removed from the `Supported` header if present) unless the Enable SIP Session Timers (SST) - callee leg action is also used.

Even if the caller does not support session timers, ABC SBC will periodically refresh the session by sending UPDATE or re-INVITE requests to the caller.

If the session timer negotiation results in the caller being responsible for session refreshes, the appropriate session refresh requests will be propagated to the callee unless the Absorb Re-INVITEs (on leg) or Absorb UPDATEs (on leg) actions are used in the caller's call leg.

Parameters

session expiration (sec)

minimum expiration (sec)

let remote refresh

1.1.59 Enable SIP Session Timers (SST) - callee leg

Enforce the use of session timers for the callee. Support for session timers is not advertised to the caller (the `timer` extension is removed from the `Supported` header if present) unless the Enable SIP Session Timers (SST) - caller leg action is also used.

Even if the callee does not support session timers, ABC SBC will periodically refresh the session by sending UPDATE or re-INVITE requests to the callee.

If the session timer negotiation results in the callee being responsible for session refreshes, the appropriate session refresh requests will be propagated to the caller unless the Absorb Re-INVITEs (on leg) or Absorb UPDATEs (on leg) actions are used in the callee's call leg.

Parameters

session expiration (sec)

minimum expiration (sec)

let remote refresh

1.1.60 Add X-Org-ConnID header

The X-Org-ConnID header field contains a unique value that remains constant for the duration of the transaction and any dialog created from this request.

By enabling this action, a X-Org-ConnID header is added to every outgoing initial SIP INVITE request product of this dialog.

The header helps to correlate calls that have been internally redirected (due to a 302 SIP response) or blindly transferred (due to a REFER SIP request).

The value can be retrieved in the CDR by specifying the keyword "$x_org_connid" in the cdr_format (see cc_syslog_cdr.conf).

1.2 SDP Mediation

1.2.1 Set CODEC Whitelist

Remove all but listed codecs from SDP.

Parameters

codec list

Comma-separated, case insensitive, list of allowed codecs.

1.2.2 Set CODEC Blacklist

Remove all listed codecs from SDP.

Parameters

codec list

Comma-separated, case insensitive, list of disallowed codecs.

1.2.3 Set CODEC Preferences

Define the order in which available codecs are chosen.

Parameters

comma-separated codec-list

1.2.4 Set SDP attribute whitelist

Removes all but listed SDP attributes from SDP payload.

Parameters

comma-separated list of attribute names

1.2.5 Set SDP attribute blacklist

Removes specified SDP attributes from SDP payload.

Parameters

comma-separated list of attribute names

1.2.6 Set SDP bandwidth limit

Set session bandwidth limit in SDP. Sets (if nonexistent) or limits (if existent) the b= attribute. If Media type is set (e.g.: 'audio' or 'video'), the media-line b= attributes are limited, if Media type is not set, the session level b= attribute is limited. Common limit types: TIAS, AS, CT.

Parameters

Limit Type

Limit type, e.g. TIAS (RFC 3890), or AS, or CT (RFC 4566)

Limit (kilobit per second)

Set media limit for this media type. Only one value is allowed.

Media type

Specify media type, e.g. 'audio' or 'video'

1.2.7 Set Media whitelist

Permit only listed media types.

Parameters

media list

Comma-separated list of enabled media types. For example “audio,video”.

1.2.8 Set Media blacklist

Remove listed media types.

Parameters

media list

Comma-separated list of media types to blacklist. For example “video,image”.

1.2.9 Drop early media

Drop early media (audio only).

1.2.10 Drop SDP from 1xx replies

Drop SDP from listed 1xx replies.

Parameters

list of affected reply codes

1.2.11 Insert or Replace SDP Session Attribute (on leg)

Try to insert a session-level attribute to all requests/replies on call leg. Unless “replace with” is enabled, the insertion will take place even if an attribute with the same name exists. If it’s enabled the value of the attribute with the same name is changed to “Attribute value”.

If the attribute is “known” to the SBC this action can remove other forms of the attribute. I.e. inserting “sendonly” will remove the previous indicator such as “inactive”, regardless of the value of the “Replace with” parameter.

Available since: 4.6.

Parameters

Attribute name

The name to replace.

Supports replacement expressions.

Attribute value

The Attribute value.

Supports replacement expressions and back-references.

Replace with

Replaces if already exists.

1.2.12 Replace SDP Session Attribute (on leg)

Replace an SDP session attribute on all requests/replies on a call leg.

Available since: 4.6.

Parameters

Attribute name

The name to replace, supports replacements.

Search

Regexp to match the part to be replaced.

Replace with

Holds the value to be replaced with. Supports replacement expressions and back-references.

1.2.13 Insert or Replace SDP Media Attribute (on leg)

Try to insert a media-level attribute to all requests/replies on call leg. Unless "replace with" is enabled, the insertion will take place even if an attribute with the same name exists. If it's enabled the value of the attribute with the same name is changed to "Attribute value".

If the attribute is "known" to the SBC this action can remove other forms of the attribute. I.e. inserting "sendonly" will remove the previous indicator such as "inactive", regardless of the value of the "Replace with" parameter.

Available since: 4.6.

Parameters

Attribute name

Name of the attribute to be replaced. Supports replacement expressions.

Media

Regexp matched against the `m=` media lines to select specific ones. Supports replacement expressions and back-references.

Attribute value

The attribute value to be used.

Supports replacement expressions and back-references.

Replace with

Replaces if already exists.

1.2.14 Replace SDP Media Attribute (on leg)

Replace an SDP media attribute on all requests/replies on a call leg.

This action can be used for payload id re-mapping if used with RTP anchor. E.g. attr. name, media, search, replace with values `rtpmap`, `.*`, `^98 XYZ`, `105 XYZ` respectively will replace payload id 98 with 105 in relayed RTP packets.

Available since: 4.6.

Parameters

Attribute name

Name of the attribute to be replaced. Supports replacement expressions.

Media

Regexp matched against the `m=` media lines to select specific ones. Supports replacement expressions and back-references.

Search

Search is a regexp to match the part to be replaced.

Replace with

Holds the value to be replaced with, supporting replacement expressions and back-references.

1.2.15 Disable SDP Media

Disable an SDP media on all requests/replies.

This action can also remove the media line based on the global config option "Remove filtered m-lines".

I.e. in removal of media with payload:

```
m=audio 8012 RTP/AVP 102
a=rtpmap:102 telephone-event/48000
a=content:special
```

"Media" would be compared against `audio 8012 RTP/AVP 102`, "Attribute name" would be compared to `rtpmap` or `content` under that media line, "Attribute value" would be compared against `102 ...` or `special` values.

Available since: 5.1.

Parameters

Media

Regexp matched against the `m=` media lines to select specific ones. Supports replacement expressions and back-references.

Attribute name

Regexp to match an attribute under the `m=` line to be removed. Supports replacement expressions and back-references.

Attribute value

Regexp to match an attribute under the `m=` line to be removed. Supports replacement expressions and back-references.

1.2.16 Remove SDP Media Attribute (on leg)

Remove an SDP media attribute on all requests/replies on a call leg.

I.e. in removal of payload with id 102:

```
m=audio 8012 RTP/AVP 102 103
a=rtpmap:102 telephone-event/48000
a=rtpmap:103 telephone-event/8000
```

"Attribute name" would be `rtpmap`, "Media" would be compared against `audio 8012 RTP/AVP 102`, "Search" would be compared to `102 telephone-event/48000`, and would result in:

```
m=audio 8012 RTP/AVP 103
a=rtpmap:103 telephone-event/8000
```

Available since: 4.6.

Parameters

Attribute name

The name of attribute to remove. Supports replacement expressions.

Media

Regexp matched against the `m=` media lines to select specific ones. Supports replacement expressions and back-references.

Search

Search is a regexp to match the line to be removed.

1.2.17 Insert or Replace SDP Payload Attribute (on leg)

Try to insert a payload-level attribute to all requests/replies on call leg. Unless "replace with" is enabled, the insertion will take place even if an attribute with the same name exists. If it's enabled the value of the attribute with the same name is changed to "Attribute value".

Available since: 4.6.

Parameters

Attribute name

The name of attribute to insert/replace. Supports replacement expressions.

Media

Regexp matched against the `m=` media lines to select specific ones. Supports replacement expressions and back-references.

Codec

Regexp matched against the respective `rtpmap=xyz <CODEC>`. Supports replacement expressions and back-references.

Attribute value

Supports replacement expressions and back-references. I.e. for `fmtp`, it is placed as `fmtp: xyz <VALUE>`.

Replace with

Replaces if already exists.

1.2.18 Replace SDP Payload Attribute (on leg)

Replace an SDP payload attribute on all requests/replies on a call leg.

Available since: 4.6.

Parameters

Attribute name

Name of the attribute to be replaced. Supports replacement expressions.

Media

Regexp matched against the `m=` media lines to select specific ones. Supports replacement expressions and back-references.

Codec

Regexp matched against the respective `rtpmap=xyz <CODEC>`. Supports replacement expressions and back-references.

Search

Regexp to match the part of attribute value to be replaced. I.e. for `fmtp` it is compared against `fmtp:xyz <SEARCH>`. Supports back-references.

Replace with

Replacement value. Supports replacement expressions and back-references.

1.2.19 Limit telephony event list (on leg)

Limit telephony events attribute on all requests/replies on a call leg.

Available since: 4.6.

Parameters

Media

Regexp matched against the `m=` media lines to select specific ones. Supports replacement expressions and back-references.

Telephony events

Comma-separated list such as `0-16,66` that will filter out anything that is not in it.

1.2.20 DTLS Setup Preference (on leg)

This controls whether SBC prefers to be *active* or *passive* for DTLS setup. I.e. when used in A-rules, if the caller signals `actpass` setup, this controls whether the SBC prefers to respond with `active` or `passive`. When used in C-rules, this can be used to configure the SBC to send `active` or `passive` instead of `actpass`.

This action is only meaningful when the RTP anchoring is in use.

Available since: 4.6.

Parameters

Preference

Can have one of the values “active”, “passive”.

1.3 Monitoring and Logging

1.3.1 Increment custom counter

Increment a custom counter. Custom counters are available via Prometheus and the legacy SNMP interface.

Parameters

counter name

increment

1.3.2 Log received traffic

Log SIP/RTP traffic concealed with logging into PCAP file.

The general log level is used if none is set for that call.

Parameters

log type

Allows choosing what to log:

- `SIP only` logs SIP messages only
- `SIP and RTP+DTMF` logs SIP messages and the whole RTP traffic.
- `SIP and DTMF only` logs SIP and only the RTP packets that are identified to be using the one of the telephone-event payload types seen in the SDP. This option is available since 5.5.

- `SIP and RTP only` logs SIP and only the RTP packets that do not have a telephone-event payload type as seen in the SDP. This option is available since 5.5.

Note that "RTP" options log all packets that are received on the RTP socket, i.e. RTCP and DTLS as well.

PCAP file name

Use filename with .pcap extension.

1.3.3 Log Event

Generate custom event

Parameters

event text

1.3.4 Set log level

Set a specific log level for this traffic.

Note: The global log level will be applied until this Action is processed.

Parameters

log level

see Section Reference of Log Level Parameters

1.3.5 Log Message

Use syslog facility.

Parameters

log level

message text

1.3.6 Log Message for Replies

Report on a transaction that completed with a specific response code. Depending on parameters, such a report can lead to blacklisting or promoting a whitelisted IP address.

Typically used to alarm on requests that were declined because of a possible security risk. The action can report via events, syslog or suggest that the request originator is put on blacklist or promoted on a greylist.

Parameters

reply codes

Comma-separated list of reply codes that trigger the reports or asterisk for any response code.

syslog level

use syslog

send an event

Blacklist UAC IP Address

Blacklist UAS IP Address

Greylist UAC IP Address

Greylist UAS IP Address

1.3.7 Log to grey list

Promote a source IP address from greylist to whitelist.

Parameters

label

Token that differentiates internally the promotion reason; choose some short descriptive string.

1.3.8 Disable privacy monitor mode

Override global configuration for privacy monitor mode to disable it for certain calls.

Note that when used in C rules, call-attempt event will still not be generated in case B-leg refuses.

Available since: 5.1.

1.4 Traffic Shaping

1.4.1 Limit parallel calls

Put a quota on number of parallel calls for some specific part of traffic identified by a key. The limit applies separately to inbound and outbound traffic in A and C rules respectively and realm or CA to which the action's rule is linked unless "global key" is turned on. Exceeding calls attempts are rejected using 403.

Parameters

max number of calls

key (optional) that identifies a subset traffic

global key

SIP header

soft limit

report abuse

SIP response code and phrase

1.4.2 Limit CAPS

Put a quota on number of call attempts per second for a traffic subset identified by a key. The limit applies separately to inbound and outbound traffic in A and C rules respectively and realm or CA to which the action's rule is linked unless "global key" is turned on. Authentication counts towards the limit as well. Exceeding calls attempts are rejected using 403.

Parameters

limit CAPS

Maximum number of request per unit of time.

time unit

length in seconds

key attribute

is global key

SIP response code

SIP response reason

SIP header

soft limit

report abuse

1.4.3 Limit Bandwidth per Call

Put a quota on RTP traffic in kbps. A rules steer bandwidth for inbound calls, C rules for outbound. Exceeding RTP traffic is dropped.

Parameters

limit (kbps)

key and global key

SIP response code and phrase

soft limit

report abuse

1.4.4 Limit Bandwidth

Don't admit signaling if its codecs in SDP exceed a limit.

Parameters

limit (kbps)

1.4.5 Set call Timer

Terminate a call if it exceeds a limit length.

Parameters

max call length

Maximum call length in seconds.

1.5 Media Processing

1.5.1 Enable RTP anchoring

Anchors RTP media to the ABC SBC.

Allows to centralize media forwarding. Anchoring is a prerequisite for other media processing such as recording.

Additionally, ICE connectivity checks and RTP keep-alive can be introduced for anchored calls. If RTP timeout is introduced and no RTP packet appears, the call is terminated.

RTCP report generation can also be configured to happen on certain conditions described in "RTCP Gen.". RTP Gen. "Always" disables RTCP relay and sends the generated RTCP (available since 4.6).

Parameters

Media far end NAT traversal

"If RFC1918 is in SDP or signaling" option for "Media far end NAT traversal" enables remote address learning only when an RFC1918 IP is seen on SDP `c=` lines or is the signaling IP for the remote endpoint in the dialog (available since 5.0).

Lock on addresses learned from RTP

Address locking affects the socket pair

"Address locking affects the socket pair" will lock both RTP and RTCP socket addresses if one of them locks before the other receives any traffic. For the socket that is locked this way, without seeing any traffic, the source port is allowed to be changed with the first packet received on that socket.

Don't send to RFC1918 addresses

Using this option will prevent the ABC SBC sending any RTP/RTCP/Other data to RFC1918 addresses on the leg.

Available since 5.0.

Enable intelligent relay (IR)

Source IP Header field for IR

Offer ICE-lite

Offer RTCP feedback

Keepalive (sec)

Timeout (sec)

Ignore ICE Offer

RTCP Generation

RTCP Interval

Change SSRC

If used ABC SBC will change the SSRC in RTP and RTCP packets with a locally generated one. Note that Convert DTMF to AVT RTP action will force-enable this behavior even if it is disabled here. For RTCP packets and SSRC replacement, only SSRC that is advertised in the SDP will get be replaced.

1.5.2 Restrict media IP to signaling IP (on leg)

Restricts the incoming and outgoing media packets to a network which is derived by applying a mask on the signaling IP address.

Packets coming from/going to a non-conforming addresses will be dropped.

Applies to RTP, RTCP and other packets.

> **Warning**
>
> This action requires RTP anchoring to be enabled as well.

Available since: 5.0.

Parameters

IPv4 Mask

"IPv4 Mask" expects a CIDR value.

"-1" means everything is allowed for IPv4 RTP.

"0" means IPv4 RTP packets will only be accepted if signaling is also IPv4 (and not v6).

"32" means packets should come from and go to the same address seen in signaling.

IPv6 Mask

"IPv6 Mask" is the IPv6 counterpart of the "IPv4 Mask" parameter.

Allow SDP IP

This option will additionally allow communication with the IP specified in respective `c=` line of the SDP.

Available since 5.2.

1.5.3 Force RTP/SRTP

Enforces conversion to the requested protocol in C-rules.

In A-rules it only admits specified protocol and declines requests otherwise. Requires RTP anchoring to be enabled.

Parameters

Key exchange mechanism (DTLS/SDES)

1.5.4 SRTP Fallback to RTP (on leg)

On the leg using this action, if a request is sent with SRTP and the remote endpoint responds with 488, the request is retried with RTP. This works for both initial INVITE and re-INVITEs / UPDATEs.

If "temporary" is false, once the leg switches to RTP, further SDP offers to it will use RTP. If it is true, then further O/A exchange will still try SRTP if it normally would (i.e. through force-srtp action or the other leg sending SRTP).

Note that if the action is on A-rules and SRTP is converted to RTP with Force RTP action on C-rules, then once a RTP-fallback occurs on A-leg, SRTP will not be retried on re-INVITEs going to a-leg even when "temporary" is set.

> **Warning**
>
> This action is only meaningful when the RTP anchoring is in use.

This action will override forcing SRTP via Force RTP/SRTP action.

Available since: 5.1.

Parameters

Temporary

1.5.5 Activate audio recording

Record audio into stereo WAV file or using a SIPREC recording server.

Recording type-specific parameters will be available based on the value of the "destination" parameter.

When WAV file recording is used, the call will be recorded as a stereo WAV file where left & right channels contain audio from A & B legs. "call-end" events will contain a link to the file holding the recording. The link will be indexed by the "audio_file" field.

When "destination" starts with `sip:`, SIPREC recording mode will be used. SIPREC-specific parameters will be available to configure options specific to the SIPREC recording mode.

Parameters

destination

Either WAV file name or SIP URI pointing to SIPREC recording server.

WAV-specific:

Discard non-established

Will discard the recording if the call ends before it is established.

SIPREC-specific:

Start announcement

ABC SBC will play an audio announcement before recording starts.

Beep tone and Beep tone interval

If set, ABC SBC will play a tone at the specified interval during the recording.

Stop announcement

ABC SBC will play an announcement before the recording stops.

Caller URI, Caller display name, Callee URI, Callee display name

These parameters are used to fill the participant fields in SIPREC metadata XML (**RFC 7865**) sent in the INVITE message to the SIPREC server.

Do not start yet

Changes the behavior to not start the recording immediately. When this option is enabled, the recording can be started when SIPREC server sends an in-dialog INFO requests with `x-ASC-Recording` header set to `started` and stopped by sending the same header with a value of `stopped`.

Stop call on SIPREC error

Stop the call when SIPREC session can not continue for any reason.

Available since 5.4.

Additional header fields

This parameter can be used to add extra headers to the messages sent to the SIPREC server.

SIP Body

This parameter can take two values. The default one is `Standard` which adds an application/rs-metadata XML in the request body. In this mode further SIPREC Extension fields can be provided. The second mode is `Custom`, which allows configuring up to 3 custom body parts with custom mime-types, headers and contents via template files.

Available since 5.4.

SIPREC Extension Data Enhancements

Adds the `<extensiondata>` section to the SIPREC metadata XML. Fields in the extension data section can be set using the respective parameters. Available in `Standard` SIP Body type.

SIPREC Extension Data | RURI

will set `<apkt:request-uri>`. Available in `Standard` SIP Body type.

SIPREC Extension Data | Realm

will set `<apkt:realm>` and `<apkt:in-realm>`. Available in `Standard` SIP Body type.

SIPREC Extension Data | Additional header fields

will be added as `<apkt:header>`. Available in `Standard` SIP Body type.

Extra Body Part | Mime Type

will add a new body part with the given mime type. Available in `Custom` body type.

Available since 5.4.

Extra Body Part | Headers

will set the new headers to the respective body part. Available in `Custom` body type.

Available since 5.4.

Extra Body Part | Body Template

will set the content of the new body part. The template engine syntax is described below. Available in `Custom` body type.

Available since 5.4.

> **Info**
>
> All header inputs can take multiple headers by separating them with `\r\n`.

Template Engine Syntax

Comments

`comments {# won't #} render` will result in `comments render`.

Loops

`loop will replaced with {% for i in range(4) %}{{ loop.index1 }}{{ i }} {% endfor %}` will result in `loop will replaced with 10 21 32 43`.

Loops can also be written using:

```
alternative
## for i in range(4)
  {{ i }}
## endfor
```

This will result in `alternative\n1\n2\n3\n4\n` where `\n` are line-feeds. In this syntax, the `##` must be at the start of the line.

Conditions

`{% for i in range(4) %}{% if loop.index1 %% 2 %}odd{% else %}even{% endif %}{% endfor %}` will result in `oddevenoddeven`.

Sorting

`sorted list is {{ sort([3,2,1]) }}` will result in `sorted list is [1,2,3]`.

List join

`hello {{ join([1,2,3], " + ") }}` will result in `hello 1 + 2 + 3`.

String manipulation

`hello {{ upper("there") }}` will result in `hello THERE`.

`hello {{ lower("THERE") }}"` will result in *hello there*.

Escaping

`{{ "{% hello %}" }}` will result in `{% hello %}`.

The SBC adds the following function extensions to the template engine:

`{{ abc_replace("<replacement-expression>") }}`

Wraps the SBC's replacement-expressions. I.e. `abc_replace("$ci")` will be replaced with the Call-ID.

`{{ abc_strftime("<format>") }}`

Converts the current system time to (UTC) to a string according to the given format. Format syntax is the same as C language's `strftime`. The final string must not exceed 127 bytes.

`{{ abc_generate_uuid() }}`

Replaced with a base64-encoded UUID.

The SBC adds the following variable extensions to the template engine:

`{{ a_leg_stream_label }}`

Replaced with the value that the SBC will put in the SDP (`a=label:<label>`) for A leg's stream.

`{{ b_leg_stream_label }}`

Replaced with the value that the SBC will put in the SDP (`a=label:<label>`) for B leg's stream.

1.5.6 Activate transcoding

Activate transcoding for list of codecs. Listed codecs are added to SDP and transcoded if selected.

When "strict SDP answer" is enabled, while sending SDP answer, SBC will only add the transcoding codecs that were in the offer. Otherwise, all the codecs in the codec list are added to the answer so that we may avoid transcoding if the UA is able to send them.

Parameters

comma-separated codec list

strict SDP answer

1.5.7 Process RTP Header Extension

Enables relaying of RTP header extension in media processor (i.e. transcoded media).

Only supports ED137A.

Available since 5.4.

1.5.8 Convert DTMF to AVT RTP

Convert detected DTMF to RTP/AVT packets (**RFC 4733**/ **RFC 2833**).

Note that this action will make the SBC replace the SSRC and sequence number in relayed RTP/ RTCP packets with locally generated ones. For RTCP packets and SSRC replacement, only SSRC that is advertised in the SDP will get be replaced.

This action can be used to convert DTMF received via SIP INFO messages or inband DTMF when used together with Activate Inband DTMF Detection action.

Parameters

Direction

Direction parameter sets on which direction to apply the conversion on.

E.g. setting it to "To B leg" on C rules would apply the conversion on DTMF generated by A leg (caller). Direction defaults to "To B leg" and "To A leg" in A and C rules respectively.

Available since 4.6.

Default volume

This parameter sets the volume when if the SBC can not figure out the volume by other means. Defaults to 20.

Available since 5.0.

Force volume

Forces the volume parameter to always be effective.

Available since 5.0.

Default duration

Sets the duration of the generated DTMF if the SBC cannot figure it out in any other means.

Available since 5.0.

Force duration

Forces the duration parameter to always be effective.

Available since 5.0.

1.5.9 Convert DTMF to SIP INFO

Same as Convert DTMF to AVT RTP except the end result is DTMF in SIP INFO messages.

When used in A rules, DTMF coming from A leg is sent as SIP INFO to B leg. When used in C rules, DTMF coming from B leg is sent as SIP INFO to A leg.

Parameters

Relay AVT RTP

This parameter can be used to control whether to drop RTP AVT packets or to also relay them.

1.5.10 Join meet-me conference

Make a call join a conference.

Note: it is strongly advised to set the configuration synchronization mode to 'pull' for nodes where 'System-generate rooms/PINs' options is enabled. A large amount of notifications about 'outdated provisioned tables' are to be expected otherwise.

Parameters

Enter room via keypad

Room

System-generated rooms/PINs

Room PINs provisioned table

Provisioned Table API user

Provisioned Table API password

Minimal room length

Unacceptable rooms

Room prefix

Split Room number and participant ID

Position to split room

Room is PIN protected

PIN

Use room's PIN as admin PIN

Record participant name

Participant recording filename

Play the number of participants in the room

Play announcements to all participants of the room

Multi-Language support (MLS)

MLS prompt directories

1.5.11 Meet-me conference set PIN

Set and persist the security PIN of a meet-me conference room into a typed provisioned table.

See Default Audio Files for more information about the defaults prompt files.

Available since: 4.6.

Parameters

Room

PIN

Source IP

Path to WAV directory

Provisioned Table API user

Provisioned Table API user password

PINs Provisioned Table

1.5.12 Refuse call with audio prompt

Play an audio announcement and decline an incoming call.

Parameters

file

The filename, relative to the global config option "Prompts/Base Directory".

As Early Media

Loop

SIP Reply and HF

1.5.13 Play prompt on final response

Play an audio announcement on receipt of a negative final response from downstream.

Parameters

SIP response codes to trigger the announcement

As Early Media

New response code if "as early media"

Optional header fields

announcement WAV filename OR characteristics of a generated ringtone

1.5.14 Generate Ring-Back Tone

Play an audio file or a dual-frequency tone instead of default ringing tone.

Parameters

On downstream 180

Start playing when a 180 response arrives.

On Timer

Start playing if a number of seconds elapses. Turned off if zero.

Generate Ringtone

If turned on, a dual-tone with specified frequencies and durations will be played; otherwise a specified audio file will be used.

File

Audio file to be played.

Loop

When audio file is chosen this option chooses whether to play it once or in a loop.

1.5.15 Activate Music On Hold

Use this action on a call to play an audio file when a call participant puts the call on hold. It is possible to specify how to signal the on-hold status in SDP.

Parameters

music file name

playback in loop

Hold indication

The method of hold signalling. Either preserve incoming or via SDP attribute (`sendonly`, `sendrecv`, `inactive`) or using connection IP set to 0.0.0.0 (**RFC 2543**).

1.5.16 Activate Inband DTMF Detection

Use this action together with the "Convert DTMF to RTP/AVT" or similar actions to detect and convert inband DTMF.

Note that this action:

- Does not filter the inband DTMF,
- will increase the CPU usage on the RTP traffic processing.

Available since: 4.6

Parameters

Direction

Can be used to set in which direction the detection will be enabled.

Mode

Can be used to i.e. not enable the detection if telephone-event is in the SDP.

1.5.17 DTMF Termination Same SSRC (on leg)

Actions that result in DTMF termination/generation (i.e. transcoding, Convert DTMF to AVT RTP) would generate the DTMF RTP (RFC4733/RFC2833) using a new SSRC. Using this action changes it to injecting DTMF RTP into ongoing RTP stream.

Note that this has the drawback of not being able to generate DTMF RTP if no other RTP packets are being relayed. This is because we can not reliably estimate RTP timestamp unless we see the live RTP traffic.

This action only acts on the packets going towards the leg it is set on. If it is in A-rules, then DTMF packets going towards the A-leg will have the same SSRC. If it is in C-rules, then DTMF packets going towards the B-leg will have the same SSRC.

Available since: 5.0

1.5.18 DTMF Termination Stable Duration Increments (on leg)

Actions that result in DTMF termination/generation (i.e. transcoding, Convert DTMF to AVT RTP) would generate the DTMF RTP (**RFC 4733**/ **RFC 2833**) using variable increments in 'duration', according to the wallclock during the relay of the other RTP packets. Using this action changes it to increment the duration in fixed steps. The step interval is determined using ptime attribute of the SDP, calculated from timestamp increments of the RTP packets or default to 20ms, in that order.

Available since: 5.2

1.5.19 Sticky Stream SSRC (on leg)

When used, the RTPs sent to the respective call agent (A-rules: Caller, C-rules: Callee) use the same SSRC value per stream. The SSRC is generated randomly for each call and is derived to be unique for each stream by using the SDP media index of the stream.

This action is only meaningful when the RTP anchoring is in use. Similar to the `Convert DTMF to AVT RTP` action, using this action will automatically change SSRC regardless of the "Change SSRC" option in RTP anchoring action, forcibly enabling it for both legs.

Available since: 5.4

1.6 SIP Dropping

1.6.1 Reply to request with reason and code

Send a response to a SIP request.

Parameters

Code

Reason

Reason phrase

Header fields

Additional header fields (optional).

Blacklist by firewall if repeated

1.6.2 Drop request

Drop request without replying.

Parameters

Event throttling key

1.6.3 Allow unsolicited NOTIFYs

Allow forwarding NOTIFY requests without a prior subscription (either implicit with REFER, or explicit with SUBSCRIBE).

1.7 Scripting

1.7.1 Set Call Variable

Stores a computing result in an variable. The variable can be tested using the Call Variable condition and/or referred to from actions using the `$V(gui.varname)` replacement.

Parameters

variable name

variable value

1.8 Register Processing

1.8.1 Enable REGISTER caching

Stores a cached copy of REGISTER contacts before forwarding.

1.8.2 Retarget R-URI from cache

Rewrites AoR in request URI with contacts cached using Enable REGISTER caching.

Parameters

enable NAT handling

enable sticky transport

1.8.3 REGISTER throttling

Force UAs to refresh registrations within a time window. Particularly useful to trigger REGISTER-based keep-alives to facilitate NAT traversal.

Parameters

minimum registrar expiration

maximum UA expiration

1.8.4 Save REGISTER contact in registrar

Act as local registrar and store registers locally.

1.8.5 Restore contract from registrar

Restore contact from registrar.

1.9 External Interaction

1.9.1 ENUM query

Make an ENUM dip. The queried value may contain replacement expression, suffix is appended to the query.

Parameters

queried value

domain suffix

ENUM services

1.9.2 Read call variables over REST

Do REST query to given URL and set call variables received in reply.

Since ABC SBC 5.3 if content-type is application/json then a json content is parsed.

Please note, neither arrays nor nested objects are supported. Only simple objects similar to the one in example are supported:

```
{
  "attribute_name": "value",
  "foo": "bar"
}
```

Parameters

REST URI

1.9.3 Read call variables from table

Read variables from a provisioned table

Parameters

table name

query key

1.10 NAT Handling

1.10.1 Enable dialog NAT handling

Remember during dialog lifetime where the initial dialog-initiating request came from and sends all subsequent SIP traffic there.

1.11 Other

1.11.1 Support serial forking proxy

Permit to reset early media upon 181-indicated serial forking.

1.11.2 Fork

Fork a new parallel branch to a URI.

Action is supported in inbound rules only.

Parameters

SIP URI

Chapter 2

Reference of Global Configuration Parameters

This reference lists all global configuration parameters used in SBC. Note that they have default values which are designated to accommodate most use-cases and can have massive impact on operation if changed: modify them only after careful consideration. The GUI screen is showing recommended default values. When the actual value is changed, the default value is highlighted as bold text.

> **Info**
>
> When the global configuration parameters are updated, a warning message with a link to activate the new SBC configuration is shown in the GUI. No changes are applied until the "activate" link is used. When the configuration changes are applied, appropriate services might be restarted (e.g. SIP and RTP processes) depending on what parameters were changed. Note that this may cause service disruption.

The configuration parameters are grouped as follows:

2.1 AWS Parameters

These parameters are used when SBC is deployed on Amazon AWS.

They are currently applied during the initial AWS configuration when using High Availability (HA) under AWS.

> **Info**
>
> Anyone in possession of an AWS IAM User Access Key can impersonate the key's owner. It is therefore recommended to create a dedicated IAM user with limited permissions and configure the SBC to access AWS using this user's identity.

For more details, see the AWS documentation on IAM user identities: https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_access-keys.html

Parameter Name	Description
Region for AWS requests	AWS region used for requests. Default value: empty API: aws_region
AWS access KEY ID	Access Key ID of an AWS user with permissions for the AWS service. Default value: empty API: aws_key_id
AWS secret access KEY	Secret key associated with the AWS user's Access Key ID. The secret is only visible when the key is created. If forgotten, the key must be regenerated. If compromised, anyone in possession of the key can impersonate the user. Default value: empty API: aws_key

Table 1: AWS Parameters

2.2 Backup Parameters

These parameters configure SBC daily backups. For more details, see Backup and Restore Operations.

Parameter Name	Description
Equivalent settings as for CCM	If enabled, the settings on this Backup tab will not be applied to SBC nodes. Instead, the same settings as configured for the CCM node (under *CCM -> CCM Config -> Backup* page) will be applied to SBC nodes. Default value: disabled API: backup_ccm_equivalent
Create daily Sbc configuration backups	If enabled, a daily snapshot of the SBC configuration will be created and stored as a gzipped tarball file. Default value: disabled API: backup_enable
Include provisioned tables in daily backups	If enabled, the daily backup will also include the content of all provisioned tables. Default value: enabled API: backup_prov

Parameter Name	Description
Number of days to keep backups	Sets the retention period for backup files. All files named 'sbc-backup-*' in the backup directory that are older than the specified number of days will be deleted during each daily backup run. Use '0' to disable automatic deletion of old backup files. Default value: 7 API: backup_keep
Destination directory for backups	Specifies the destination directory for the daily backup files. Default value: /data/backups API: backup_dir
Full path to extra files or dirs to include in backup	Extra custom files or directories to be included in the backup, using full paths. Multiple entries can be separated by commas. A '*' wildcard may be used. Paths must not contain commas. Default value: empty API: backup_extra_files

Table 2: Backup Parameters

2.3 CDR Parameters

These parameters define how and where CDRs are stored. For more details, see Call Data Records (CDRs).

Parameter Name	Description
Enable CDRs	Enables writing of Call Detail Records (CDRs). **Changing this value may restart signaling process.** Default value: enabled API: cdr_enabled
Number of daily CDR files to keep	CDR retention policy. The SBC produces CDRs for all completed calls in CSV format. This setting specifies the number of CDR files to retain. **Changing this value may restart signaling process.** Default value: 93 API: cdr_days_keep
Directory for exported CDR files	Directory in the filesystem where the CSV CDR files are stored. Default value: /data/cdr/export API: cdr_export_dir

Parameter Name	Description
CDR files rotation frequency	Sets the frequency of CDR file rotation. Supported values: “daily”, “weekly”, or “monthly”. The number of rotated files to keep before deletion is defined by the *Number of CDR files to keep* parameter. Default value: daily API: cdr_rotate_freq Possible values over API: • hourly • daily • weekly • monthly • yearly
Enable new version of CDRs (CDR-NG)	Enables the new version of CDRs, called CDR-NG. This feature is in an experimental state. **Changing this value may restart signaling process.** Default value: disabled API: cdr_ng_enabled
CDR format	Defines the structure of CDR records. See Customized CDR Records. **Changing this value may restart signaling process.** Available since: 5.6 Default value: $srcrlm.name, $srcca.name, $dstrlm.name, $dstca.name, caller_id_user, caller_id_host, caller_id_name, callee_id_user, callee_id_host, callee_id_name, $ltag, $start_tm, $connect_tm, $end_tm, $duration, $setup_duration, $bill_duration, sip_req_uri, sip_from_uri, sip_to_uri API: cdr_legacy_fmt

Table 3: CDR Parameters

2.4 Conference Parameters

This section contains the configuration parameters for meet-me Conference feature.

Parameter Name	Description
Keep participant’s name recordings for	Duration (in hours) for which files containing web conference participants’ names are retained on the file system. These files are not subject to replication. **Changing this value may restart signaling process.** Default value: 90 API: wc_preserve_participants_files_for

Parameter Name	Description
Play the number of participants in the room	If enabled, the number of participants is announced when a participant joins or leaves the conference room. Alternatively, a participant can press the star (*) key during a call to hear this information. **Changing this value may restart signaling process.** Default value: disabled API: wc_echo_nb_participants
Use room security pin value for the admin pin	If both *Use security pin* and *Use admin pin* options are enabled for a room, the admin pin value is set to the same value as the security pin. **Changing this value may restart signaling process.** Default value: disabled API: wc_use_secu_as_admin_pin
Path to directory holding digits wav files	Path to the directory containing audio files used to announce numbers (e.g., "one", "twenty", "(seven-)teen"). By default, SBC ships two versions of this directory: `/usr/lib/sems/audio/webconference/digits/` for English prompts, and `/usr/lib/sems/audio/webconference/de/digits/` for German prompts. **Changing this value may restart signaling process.** Default value: /usr/lib/sems/audio/webconference/digits/ API: wc_path_to_digit_wav
List of provisioned tables to watch for expired room	List of provisioned tables to watch for expired rooms. Generated web conference names and PINs are stored in the CCM provisioned tables. The CCM attempts to remove expired rooms from these tables every day at 02:00. Default value: empty API: wc_generated_table_to_watch
Generated rooms validity	Number of days a generated conference room remains active. Once closed, the room's PIN is blocked for a fixed period. **Changing this value may restart signaling process.** Default value: 30 API: wc_block_generated_room_after
Keep expired generated rooms	Number of days before a closed generated conference room's PIN is unblocked. **Changing this value may restart signaling process.** Default value: 90 API: wc_block_generated_room_for

Table 4: Conference Parameters

2.5 Eventbeat Parameters

These parameters are used to tweak and debug event communications between an SBC node and an ABC Monitor node. They also enable generating statistics, which can be exposed through the application interface on TCP ports :4247 and :4248.

Parameter Name	Description
Event batching size	Maximum number of events sent at once to the monitor. Default value: 10 API: eventbeat_batch
Enable eventbeat statistic reporting	Exposes live event processing metrics on a TCP port (`:4247` for the primary ABC Monitor and `:4248` for the secondary ABC Monitor). Default value: enabled API: eventbeat_enable_stats
Interval between each statistic	Interval in seconds at which a single statistics entry is recorded. Default value: 5 API: eventbeat_stat_interval
How many statistic entries per payload	Number of statistics entries to be returned in a single payload. Example: To collect statistics for the last minute in 5-second intervals: • Set *Interval between each statistic* to `5` • Set *How many entries per payload* to `12` Default value: 10 API: eventbeat_stat_report

Table 5: Eventbeat Parameters

2.6 Event Parameters

These parameters are used to define how and where events are stored.

Parameter Name	Description
Number of days to keep old traffic log files	Defines the local retention policy for traffic log files. Particularly useful when no ABC Monitor is attached to the SBC. Must be shorter than the retention policy on ABC Monitor to avoid copying files already expired on ABC Monitor. Default value: 7 API: events_days_keep
ABC Monitor address	Specifies the IP address or DNS name of ABC Monitor. Empty if no ABC Monitor is attached to the SBC. **Changing this value may restart signaling process.** Default value: empty API: events_remote_console
Secondary ABC Monitor address	Specifies the IP address or DNS name of the secondary ABC Monitor. Empty if no secondary ABC Monitor is attached to the SBC. **Changing this value may restart signaling process.** Default value: empty API: events_remote_console2

Parameter Name	Description
Replicate traffic logs to ABC Monitor	If enabled, collected PCAP files are sent to a Monitor server using the rsync protocol. Files are deleted from the SBC after transfer. Default value: enabled API: replicate_traffic_log
Replicate traffic logs to secondary ABC Monitor	If enabled, collected PCAP files are sent to the secondary Monitor server using the rsync protocol. Files are deleted from the SBC after transfer. Default value: enabled API: replicate_traffic_log2
Traffic Log Chunk Seconds	The duration at which the PCAP is split into chunks. Default value: 10 API: traffic_log_chunk_secs
Replicate recordings to ABC Monitor	If enabled, recorded audio files (see Section Audio Recording) are sent to a Monitor server using the rsync protocol. Files are deleted from the SBC after transfer. Default value: enabled API: replicate_recordings
Replicate recordings to secondary ABC Monitor	If enabled, recorded audio files (see Section Audio Recording) are sent to the secondary Monitor server using the rsync protocol. Files are deleted from the SBC after transfer. Default value: enabled API: replicate_recordings2
Replication rsync password	Specifies the rsync password used for replicating traffic logs and recorded audio files. Available up to: 5.5 Default value: empty API: replication_rsync_password
Replication rsync password for secondary ABC Monitor	Specifies the rsync password used for replicating traffic logs and recorded audio files to the secondary Monitor. Available up to: 5.5 Default value: empty API: replication_rsync_password2
Use TLS secure connection to ABC Monitor	If enabled, events, traffic logs, and recording files are sent to ABC Monitor over a TLS-secured connection. It is recommended to install a trusted certificate on ABC Monitor instead of the default self-signed certificate. On the SBC side, the TLS profile of the IMI interface is used. Default value: disabled API: events_use_tls
Number of hours to keep old recordings (0 to not delete)	Defines the retention policy for recorded WAV files. A value of 0 disables deletion. Default value: 168 API: recordings_hours_keep

Parameter Name	Description
Generate an event if a SIP transaction reaches the defined number of retransmissions	Generates a “notice” event when the configured number of SIP retransmissions is reached. These events appear in the ABC Monitor Transport Dashboard. Use with care — low values may cause excessive events. If used, recommended value is 4. Use ‘0’ to disable this functionality. **Changing this value may restart signaling process.** Default value: 0 API: sip_retrans_event_trigger
Maximum number of events buffered in local Redis	Defines the retention policy for locally buffered events in Redis. Use ‘0’ for no limit. **Changing this value may restart signaling process.** Default value: 0 API: redis_max_local_events
List of call variables added into events	Specifies a list of call variables to include in call events. Format: `<var_name>:<flag>`, where `<flag>` is 0 or 1, indicating whether the value can be overwritten. Use * as a wildcard to include all. **Changing this value may restart signaling process.** Default value: empty API: cvars_allowed
Generate an event on UDP receive buffer errors	If enabled, an alert is generated when UDP receive buffer errors are detected on the system network interface. Default value: enabled API: checknet_udp_recv_err
Generate an event on UDP send buffer errors	If enabled, an alert is generated when UDP send buffer errors are detected on the system network interface. Default value: enabled API: checknet_udp_send_err
Generate an event on UDP packet receive errors	If enabled, an alert is generated when UDP packet receive errors are detected on the system network interface. Default value: enabled API: checknet_udp_pacerr_err
Generate an event on IP incoming packet receive errors	If enabled, an alert is generated when IP incoming packet receive errors are detected on the system network interface. Default value: enabled API: checknet_ip_incdis_err
Generate an event on outgoing packets dropped errors	If enabled, an alert is generated when outgoing packets dropped errors are detected on the system network interface. Default value: enabled API: checknet_ip_outdrop_err

Parameter Name	Description
Alarm when number of calls reaches % of the license	Issues a warning when the number of sessions reaches X% of the license limit. An info-level event is generated when the session count drops back below X%. **Changing this value may restart signaling process.** Default value: 75 API: session_nearly_limit_percent
Privacy monitor mode	If enabled, prevents sending call-attempt, call-start, and call-end events to ABC Monitor. Can be overridden using the "Disable privacy monitor mode" action. **Changing this value may restart signaling process.** Default value: disabled API: privacy_monitor
Destination monitor event interval (sec)	Sets the interval, in seconds, at which destination monitor events are generated. **Changing this value may restart signaling process.** Default value: 300 API: interval_dest_monit
Threshold of number of events buffered on Sbc to set warning	Sets the maximum number of events allowed in the Redis queue on the SBC before the node status is set to "warning" on the System Monitoring page. Default value: 500 API: events_buffer_treshold
Enable events redis disk persistence	If enabled, enables Redis disk persistence for events, using the `/data/redis` directory. Ensure sufficient disk space before enabling. Default value: disabled API: events_redis_persistence
Periodic RTP Statistics	Enables sending periodic RTP statistics per call leg at 10-second intervals. **Changing this value may restart signaling process.** Available since: 5.4 Default value: disabled API: periodic_rtp_stats

Table 6: Event Parameters

2.7 Firewall Parameters

This section contains the configuration parameters for Firewall.

Parameter Name	Description
Enable SBC firewall	If enabled, the firewall chains are populated with SBC firewall rules. If deployed in a container or on a system not supporting nftables or nfsets, disable this option to avoid SBC node errors in System status. Note: on SBC < 5.4 the firewall uses iptables and ipsets. **Changing this value may restart signaling process.** Available since: 5.0 Default value: enabled API: fw_enable
Reject packets instead of dropping	If enabled, packets not allowed by the firewall are rejected and an ICMP admin-prohibited message is sent. If disabled, packets are silently dropped. Note: on SBC < 5.4 the firewall uses reject; starting with 5.4 it uses drop by default. Available since: 5.4 Default value: disabled API: fw_reject
Do not accept any ICMP packets	If enabled, incoming ICMP packets are dropped and not answered. Use with caution, as blocking ICMP may cause connectivity issues. Available since: 5.5 Default value: disabled API: fw_block_icmp
Blacklist IP addr for repeated signaling failures	If enabled, IP address of request that failed authentication, exceeded limit, failed sanity check, was dropped by “Drop” action or “Log message / Event for replies” action was used, will be added to the blacklist, silently dropping all packets from it. Note: the specific blacklist reasons must also be enabled in CA settings or in the “Drop” or “Log message/Event for replies” actions. See Section Automatic IP Address Blocking for details. Default value: disabled API: fbl_sig_enable
Signaling failures blacklist: IP address start score before any offense	Sets the starting score used before any offenses are registered. This value decreases with each offense until it reaches 0 or less, at which point the IP address is blacklisted. See Section Automatic IP Address Blocking for more details. **Changing this value may restart signaling process.** Default value: 2.8 API: fbl_start_score

Parameter Name	Description
Signaling failures blacklist: rate per second used to calculate a time-related bonus between offenses	Sets the allowed rate of offenses in events per second. Allows the score to recover slightly over time, acting as a bonus for good behavior. See Section Automatic IP Address Blocking for more details. **Changing this value may restart signaling process.** Default value: 0.0005 API: fbl_allowed_rate
Signaling failures blacklist: time in seconds to remove entries for which no event has occurred from score calculation	Sets the number of seconds after which, if no offense is seen from an IP address, it is removed from the scoring table. New offenses from removed IPs use the start score. See Section Automatic IP Address Blocking for more details. **Changing this value may restart signaling process.** Default value: 600 API: fbl_inactive_timeout
Time in seconds to blacklist IP addr for signaling failures	Sets how long an IP address remains blacklisted for signaling failures before automatic removal (for drop, failed auth, limit, sanity). See Section Automatic IP Address Blocking for more details. Default value: 3600 API: fbl_sig_bantime
Greylist: time delay in seconds to give IP a chance to prove validity	If the traffic from IP address proves validity during this probation period, the source IP addr will be added to whitelist. Note that the corresponding action options like “Greylist IP address” or “Log to greylist” have to be used. See Section Automatic Proactive Blocking: Greylisting for more details. Default value: 600 (seconds) API: fgl_delaytime
Greylist: time period in seconds when IP can be blacklisted if repeats and did not prove validity	If traffic from IP address did not prove validity during the probation time period, and new packet comes during this time period since first packet, the source IP addr will be added to blacklist. Note that the “Greylist” flag has to be enabled on SBC signaling interface for this to work. All traffic from the IP addresses on blacklist will be silently dropped. See Section Automatic Proactive Blocking: Greylisting for more details. Default value: 1800 (seconds) API: fgl_learntime
Greylist: time in seconds to keep IP on blacklist	Sets how long to keep the IP address on blacklist. After this time it is removed from blacklist and has a chance to prove validity again. See Section Automatic Proactive Blocking: Greylisting for details. Default value: 86400 (seconds) API: fgl_bltime

Parameter Name	Description
Greylist: time in seconds to keep IP on whitelist	Sets how long to keep IP address on whitelist. After this time it is removed from whitelist and has to prove validity again. See Section Automatic Proactive Blocking: Greylisting for details. Default value: 86400 (seconds) API: fgl_wltime
Greylist: additional ports or port ranges (a:b) to check in addition to signaling ports, space separated	Sets additional ports to ports defined on SBC signaling interfaces. If used, traffic coming to this port(s) will be also subject to the greylisting procedure. You can specify single port(s) or port ranges (in format lower:higher), space separated. See Section Automatic Proactive Blocking: Greylisting for details. Default value: empty API: fgl_extraports
Log blacklisted IP addresses to syslog	If enabled, logs blacklisted IP addresses to syslog. Entries are stored in `/var/log/frafos/sems-blacklist.log`. **Changing this value may restart signaling process.** Default value: disabled API: fbl_syslog_enable
Log greylisted IP addresses to syslog	If enabled, logs greylisted IP addresses to syslog. Entries are stored in `/var/log/frafos/sems-greylist.log`. **Changing this value may restart signaling process.** Default value: disabled API: fgl_syslog_enable
Overall limit in packets per second from not approved IP addresses	This option can be used to set overall packets per second limit on all IP addresses, that did not prove validity using “Greylist IP address” or “Log to greylist” action options. Use with caution. Use ‘0’ to disable any rate limiting. Default value: 0 API: ratelimit_notwl
Filter incoming signaling and media traffic by input interface name	Allow incoming signaling and media traffic only from corresponding interfaces. It may be needed to be turned off in some cases like more interfaces connected to the same segment or macvlan interfaces used, depending also on ARP related settings. Available since: 5.5

Table 7: Firewall Parameters

2.8 Lawful Interception Parameters

This section contains the configuration parameters for Lawful Interception (LI).

Parameter Name	Description
Lawful Interception enabled	If enabled, activates the lawful interception feature. This must also be configured under the corresponding action to take effect. **Changing this value may restart signaling process.** Default value: disabled API: li_enabled
Operator ID	Specifies the Operator ID value. **Changing this value may restart signaling process.** Default value: empty API: li_operator_id
Delivery Country Code (DCC)	Specifies the Delivery Country Code (DCC) value. **Changing this value may restart signaling process.** Default value: empty API: li_deliv_cc
Network Element Identifier	Specifies the Network Identifier (NID), which consists of the operator ID and, optionally, the network element. **Changing this value may restart signaling process.** Default value: empty API: li_network_element_id
Interception Point ID	Specifies the Interception Point ID value. **Changing this value may restart signaling process.** Default value: empty API: li_interception_point_id

Table 8: Lawful Interception Parameters

2.9 Low-level Parameters

These settings take effect only after a server reboot. Additional information can be found in Hardware Specific Configurations.

> ⚠ **Warning**
>
> Changing these parameters may significantly alter system behavior. Their impact largely depends on the equipment in use.

Parameter Name	Description
Interfaces where to enable RPS	Network interfaces on which a "receive packet steering" kernel feature should be enabled, separated by spaces. While the kernel leaves this option by default off, turning it on can increase media throughput. Default value: empty API: tunings_rps_interfaces

Parameter Name	Description
Interfaces where to set ethtool options	Network interfaces, separated by spaces, where to apply coalesce and ringbuffer *ethtool* options specified in "Coalesce ethtool options" and "Ringbuffer ethtool options" attributes. Default value: empty API: tunings_ethtool_interfaces
Coalesce ethtool options	Ethernet adapter coalescing options (syntax of *ethtool*). Applied to interfaces listed in "Interfaces where to set ethtool options". Tunes the trade-off between less-CPU-intensive and more-real-time packet processing latency; results depend on used network card. Default value: rx-usecs 200 tx-usecs 100 API: tunings_ethtool_coalesce
Ringbuffer ethtool options	Ethernet adapter RX/TX ring parameters (syntax of *ethtool*). Applied to listed interfaces in "Interfaces where to set ethtool options". Fine-tuning this parameter is specific to used network card. Increasing buffer sizes allows to deal with temporary packet bursts, while latency may increase. Default value: rx 1000 tx 1000 API: tunings_ethtool_ringbuffer
Interfaces where to bind irqs to CPUs	Network interfaces on which the individual interrupts for receive and transmit queues should be statically bound to individual CPUs / CPU cores. This option may increase media throughput on network cards with multiple queues. Default value: empty API: tunings_irqs_interfaces
Run db check on boot	If enabled, runs `mysqlcheck` during boot to help recover from unexpected shutdowns. May slow startup. Default value: enabled API: tunings_run_mysqlcheck
Clean tmp files on boot	If enabled, clean-up system directory for temporary files at boot. Default value: enabled API: tunings_clean_tmp
Sems memory limit in % from total memory	Limits maximum memory usage of the Sems process as a percentage of total system memory. Set to '0' for no limit. Default value: 75 API: sems_mem_limit
Provisioned tables redis disk persistence time interval (in seconds)	Time interval (seconds) after which provisioned tables data on an SBC backup node is saved from Redis to disk for persistence. Saving occurs only if changes exceed the next setting's threshold. Default value: 600 API: redis_prov_savetime

Parameter Name	Description
Provisioned tables redis disk persistence number of records to trigger save	Minimum number of provisioned tables records changes that trigger save to disk. The data will be saved when both the number of changed records and the time interval conditions are met. Default value: 1 API: redis_prov_saverecords
Use real-time priority on provisioned tables redis	If enabled, real-time process priority will be used on provisioned tables redis db, which helps performance. It can only be used when operating system or container permissions support this. For podman installations please make sure the `--cap-add=CAP_SYS_NICE` is used if redis real-time priority is required. Default value: disabled API: redis_prov_enable_rt
Session processor threads	Threads that process the SIP signaling and the B (routing) and C (outbound) rules of the ABC rule set. They are created in a thread pool among which all SIP sessions are distributed. Recommended: CPU threads x 2, minimum 8 threads. If the SBC needs to process a lot of external data in the routing or C rules, e.g. needs to query provisioned tables or external API server via REST, then it is recommended to set this to a high number. **Changing this value may restart signaling process.** Default value: 16 API: sems_session_processor_threads
Media processor threads	Threads for RTP transcoding and media apps (conferencing, announcements). In normal SBC operation, when those functionalities are not used, these threads will be idle. Like the session processor threads, the number configured here sets the number of threads created in a thread pool among which all media sessions are distributed. If transcoding or media applications are used, it is recommended to set this number to CPU threads x 2, otherwise it is recommended to leave them to the default or even less. **Changing this value may restart signaling process.** Default value: 16 API: sems_media_processor_threads

Parameter Name	Description
SIP server threads	These threads receive SIP messages from the network and initially parse them for later processing by the Session processor threads, immediately reply e.g. if the reply is given by the SIP dialog state (e.g. errors). They also process the A rules of the ABC rule sets. The number of threads configured here is started for every signaling interface (SI), and one set for udp and one for TCP; so e.g. if five SI interfaces are configured, and this is set to 4, then 5*4*2=40 threads are started. The recommended number depends on the number of signaling interfaces; e.g. on a setup with two signaling interfaces, the recommended number would be equal to the number of CPU cores (e.g. 8, 16 or 32). On a setup with many signaling interfaces, this should be set to e.g. 2 or 4. **Changing this value may restart signaling process.** Default value: 16 API: sems_sip_server_threads
Out-of-dialog requests threads	Threads handling REGISTER, SUBSCRIBE/NOTIFY and MESSAGE requests. It is recommended to increase this number, if processing many registrations or subscriptions. **Changing this value may restart signaling process.** Default value: 1 API: sems_ood_requests_threads
RTP receiver threads	Threads that receive and relay RTP (including SRTP decrypt when enabled). As with the thread pools above, this number is a global number of threads for a thread pool. The recommended number to set this to is 2-4 times the usable CPU hardware threads. **Changing this value may restart signaling process.** Default value: 16 API: sems_rtp_receiver_threads
Call restore threads (HA)	Thread pool used only for call restoration after HA failover. It is recommended to set it to the number of usable CPU hardware threads. **Changing this value may restart signaling process.** Default value: 4 API: restore_threads
HA interval to send adverts to peer, in seconds	Keepalived advert interval (seconds). Lower values shorten failover detection time but may cause stability issues. It is recommended keeping the default value for typical setup. Default value: 1 API: ha_advert_int
HA priority	VRRP priority (1-255). The `255` has special meaning of address owner. It is recommended to keep at default value for typical deployments. Default value: 250 API: ha_priority

Parameter Name	Description
HA mode	Configures the HA mode to use. This option should be modified only if required by a specific setup. **Warning:** Changing this option may trigger an HA switchover. The default mode uses multicast on the IMI interface (or a custom interface where the HA application is enabled) for HA VRRP advertisements. The second mode uses unicast on the IMI interface (or a custom interface where the HA application is enabled) for HA VRRP advertisements. The third mode uses multicast on both IMI and SIG interfaces for HA VRRP advertisements, providing greater robustness. For this mode to work, the system interface corresponding to the signaling interface must also have a non-VIP IP address. The fourth mode uses multicast only on SIG interfaces for HA VRRP advertisements and employs macvlan sub-interfaces for VIP IP addresses. This mode is typically not needed, but it can improve switchover performance in some cases. For this mode to work, the system interface corresponding to the signaling interface must also have a non-VIP IP address. In some cases, it may be needed to turn off the global config option "Filter incoming signaling and media traffic by input interface name" uder Firewall tab. Available since: 5.5 Default value: use multicast, vrrp on IMI interface only API: ha_mode Possible values over API: • "default": "use multicast, vrrp on IMI interface only" • "unicast": "use unicast, vrrp on IMI interface only" • "imisig": "use multicast, IMI and SIG interfaces for vrrp" • "vmac": "use multicast, use only SIG interfaces for vrrp and use macvlan for VIPs"

Table 9: Low-level Parameters

2.10 Miscellaneous Parameters

This section contains miscellaneous configuration parameters, including GeoIP and other general-purpose settings.

Parameter Name	Description
Blacklist timeout for IP addresses from external sources	Timeout (seconds) for IP addresses blacklisted by RESTful requests. Default value: 86400 API: externalbl_timeout
Enable sending important syslog entries to ABC monitor	If enabled, sends syslog entries with severity `critical` through `emergency` as alerts to the ABC Monitor. Default value: disabled API: enable_syslog_event

Parameter Name	Description
Session Management enable	If enabled, advanced load balancing is used. **Changing this value may restart signaling process.** Default value: disabled API: session_management_enable
GeoIP - account ID for geoipupdate command	Account ID used with the `geoipupdate` command, which periodically updates the GeoLite2 database if a license is provided. The license must be created with a MaxMind account by user. Default value: empty API: geoip_account_id
GeoIP - license key for geoipupdate command	License key used with the `geoipupdate` command, which periodically updates the GeoLite2 database if a license is provided. The license must be created with a MaxMind account by user. Default value: API: geoip_license_key

Table 10: Miscellaneous Parameters

2.11 PCAP Parameters

These parameters define how the most recent SIP traffic is recorded on the system for troubleshooting purposes. The SBC stores SIP traffic in PCAP files of a configured size and automatically deletes the oldest files when the retention limit is reached. The recorded PCAP files can be accessed and inspected in the administrative interface, as described in Section User Recent Traffic.

Parameter Name	Description
File size in MB for one pcap file	Maximum size of a PCAP file before a new file is created. Default value: 50 API: pcap_file_size
Number of pcap files to keep	Defines the PCAP retention policy. Files are rotated, and only the configured number of files is kept. Least recent files are deleted. Use 0 to disable SIP traffic capture (not recommended). Note: PCAP filenames use ".pcapXX", where XX is the file number. Changing the number of files deletes all existing traffic.pcap* files once configuration changes are activated. Default value: 0 API: pcap_file_count

Table 11: PCAP Parameters

2.12 RTP handling Parameters

This section contains configuration parameters that control how RTP streams are handled and processed.

Parameter Name	Description
Drop unnegotiated RTP payloads	If enabled, RTP packets with payload types not negotiated in the SDP are dropped. **Changing this value may restart signaling process.** Default value: disabled API: drop_unnegotiated_payloads
Do not drop unnegotiated comfort noise RTP payload	If enabled, RTP packets with static comfort noise payload type (13) are not dropped even if not negotiated in SDP and *Drop unnegotiated RTP payloads* is enabled. **Changing this value may restart signaling process.** Default value: disabled Available since: 5.3 API: dont_drop_unnegotiated_cn
Force symmetric RTP for media server apps	If enabled, embedded media processing actions ignore IP addresses in the caller's SDP and send RTP to the source of received RTP packets. **Changing this value may restart signaling process.** Default value: enabled API: sems_force_symmetric_rtp
RTP keep-alive frequency	Defines how often (in seconds) the SBC sends RTP keep-alive packets to its peers. See Setting RTP Inactivity Timer and Keepalive Timer. **Changing this value may restart signaling process.** Default value: 0 API: sems_rtp_keepalive_freq
RTP keep-alive method	Defines which RTP keep-alive method is used. **Changing this value may restart signaling process.** Default value: "RTP version equals 0" API: sems_rtp_keepalive_method Possible values over API: • "0": RTP version equals 0 • "1": All RTP header fields equal 0
RTP timeout	Defines the time (in seconds) after which a call is terminated if RTP packets stop arriving. See Setting RTP Inactivity Timer and Keepalive Timer. **Changing this value may restart signaling process.** Default value: 0 API: sems_rtp_timeout

Parameter Name	Description
Learn remote media address interval	Interval (ms) after the first RTP packet is received during which the remote address may still change and be re-learned. After this interval SEMS locks the remote address. Especially for re-learning after re-Invite, this may prevent locking on the old address due to some late RTP packets from the old remote address. Use 0 to disable (lock on the first packet). **Changing this value may restart signaling process.** Default value: 1000 API: learn_raddress_interval
Recording playout buffer type	Defines the playout buffer type used when recording WAV files. Possible values: • adaptive: Recommended, especially for higher jitter and packet loss showing in the RTP stream. • simple: Basic buffer, may fail on lossy links. **Changing this value may restart signaling process.** Default value: adaptive API: recording_playout_buffer Possible values over API: • adaptive • simple
MOS Average S Coefficient	Coefficient for short-term MOS calculation. The value is updated with the formula `value=value*coefficient+new_value*(1-coefficient)` on each call end where new_value is the MOS average on call end. Initial value for `value` is `4.409405954387269`. **Changing this value may restart signaling process.** Default value: empty API: rtp_mos_average_s
MOS Average M Coefficient	Coefficient for medium-term MOS calculation. The value is updated with the formula `value=value*coefficient+new_value*(1-coefficient)` on each call end where new_value is the MOS average on call end. Initial value for `value` is `4.409405954387269`. **Changing this value may restart signaling process.** Default value: empty API: rtp_mos_average_m
MOS Average L Coefficient	Coefficient for long-term MOS calculation. The value is updated with the formula `value=value*coefficient+new_value*(1-coefficient)` on each call end where new_value is the MOS average on call end. Initial value for `value` is `4.409405954387269`. **Changing this value may restart signaling process.** Default value: empty API: rtp_mos_average_l

Table 12: RTP handling Parameters

2.13 SEMS Parameters

These parameters define the behavior of the ABC-SBC engine, specifically the SEMS signaling and media processor. They are primarily intended for troubleshooting and performance tuning, and should only be modified when necessary.

Parameter Name	Description
Use raw sockets	Performance optimization technique for sending RTP packets on Linux systems with a slow UDP stack. **Changing this value may restart signaling process.** Default value: disabled API: use_raw_sockets
Default Destination Blacklist TTL	Defines how long (ms) unavailable IP destinations are maintained on a blacklist to which no SIP traffic is sent by default. For Call Agent, a specific value may be entered in the CA parameters. See IP Blacklisting: Adaptive Availability Management. **Changing this value may restart signaling process.** Default value: 0 API: default_bl_ttl
Load q850_reason call control module	If enabled, the module for processing Q.850 reasons will be loaded. The file cc_q850_reason.conf is empty by default and must be provided as a custom local template (/data/local-templates/etc/sems/cc_q850_reason.conf). **Changing this value may restart signaling process.** Default value: disabled API: q850_reason
General RT priority	General thread real-time priority value. **Changing this value may restart signaling process.** Default value: 5 API: sems_rt_prio
SIP thread RT priority	Real-time priority for SIP threads. **Changing this value may restart signaling process.** Default value: 50 API: sems_sip_server_rt_prio
RTP RT priority	Real-time priority for RTP threads. **Changing this value may restart signaling process.** Default value: 90 API: sems_rtp_receiver_rt_prio
Session processor RT priority	Real-time priority for session processor threads. **Changing this value may restart signaling process.** Default value: 50 API: sems_session_processor_rt_prio

Parameter Name	Description
Media processor RT priority	Real-time priority for media processor threads. **Changing this value may restart signaling process.** Default value: 90 API: sems_media_processor_rt_prio
Websocket ping-pong interval in seconds	Interval in seconds for keepalive ping-pong messages on WebSocket signaling interfaces. Use 0 to disable. **Changing this value may restart signaling process.** Default value: 20 API: ws_ping_pong
Soft limit for out-of-dialog transactions (event logging only)	Maximum number of out-of-dialog (OOD) server transactions before triggering an alert event. Applies when creating a new OOD transaction not linked to an existing dialog. Use 0 to disable. **Changing this value may restart signaling process.** Default value: 0 API: tr_soft_limit_ood
Hard limit for out-of-dialog transactions (event + reply 503)	Maximum number of active server transactions not related to existing dialogs. When exceeded, new requests are rejected with "503 Overloaded" and a monitoring event is generated. Use 0 to disable. See section Server Transaction limits for details. **Changing this value may restart signaling process.** Default value: 0 API name: tr_hard_limit_ood
Event throttling for soft/hard OOD limit	Limits the number of events generated by soft/hard out-of-dialog transaction limits to one per type (soft/hard) per configured time interval (in seconds). Use 0 to disable. See section Server Transaction limits for details. **Changing this value may restart signaling process.** Default value: 0 API name: tr_limit_ood_throttle
Soft limit for in-dialog transactions (event logging only)	Number of active in-dialog server transactions that, if exceeded, will trigger an alert event. Applies only to transactions tied to an existing dialog. Use 0 to disable. See section Server Transaction limits for details. **Changing this value may restart signaling process.** Default value: 0 API name: tr_soft_limit_dlg

Parameter Name	Description
Hard limit for in-dialog transactions (event + reply 503)	Maximum number of in-dialog server transactions allowed. When exceeded, new requests are rejected with "503 Overloaded" and a monitoring event is generated. Use 0 to disable. See section Server Transaction limits for details. **Changing this value may restart signaling process.** Default value: 0 API name: tr_hard_limit_dlg
Event throttling for soft/hard DLG limit	Limits the number of events generated by soft/hard DLG transaction limits to one per type (soft/hard) per configured time interval (in seconds). Use 0 to disable. See section Server Transaction limits for details. **Changing this value may restart signaling process.** Default value: 0 API name: tr_limit_dlg_throttle
Loop detection secret	Used to create a special branch tag. When a new request is received containing this tag in the first Via-HF, it is rejected with "482 Loop Detected". Accepted values: • empty: disables the feature • auto: generates an automatic secret string • string: use provided string **Changing this value may restart signaling process.** Default value: auto API name: loop_detection_secret
Strict URI user charset check	If enabled, the user part of a URI is restricted to the characters defined by RFC3261 (ABNF rules). If disabled, SBC allows more characters as long as URI parsing is not broken. **Changing this value may restart signaling process.** Default value: enabled API name: strict_uri_user_charset
TCP connection idle timeout in milliseconds	Sets the TCP connection idle timeout (in ms) on signaling interfaces. Use 0 to disable. **Changing this value may restart signaling process.** Default value: 0 API name: tcp_idle_timeout
TCP send timeout for signaling interfaces	Corresponds to TCP_USER_TIMEOUT from tcp(7). **Changing this value may restart signaling process.** Default value: 20000 API name: tcp_user_timeout

Parameter Name	Description
Delay after startup to ignore limits	Delay (in seconds) during which CAPS and other limits are ignored after SBC startup. **Changing this value may restart signaling process.** Default value: 30 API name: inhibit_caps_timer
RESTful interface - verify https peer	If enabled, HTTPS peer certificate validity is checked in RESTful queries. **Changing this value may restart signaling process.** Default value: enabled API name: rest_verify_https_peer
Custom CA path for request queries	Custom CA certificate path for SEMS REST module. If provided, every outgoing HTTPS request uses this CA. SBC automatically adds the CA to the trusted chain. Process: • Copy CA to `/etc/pki/ca-trust/source/anchors/` • Run `update-ca-trust` **Changing this value may restart signaling process.** Default value: empty API name: rest_ca_file
User-agent string	Sets the SIP User-Agent header value if specified. **Changing this value may restart signaling process.** Default value: empty API name: user_agent_string
Unprocessed events limit	Maximum size of Redis write queue when Redis is offline. If exceeded, new writes are ignored. Set 0 to disable limit. **Changing this value may restart signaling process.** Default value: 20000 API name: redis_write_queue_limit
Unprocessed events limit warning threshold	Threshold of queued writes (see `Unprocessed events limit` option) at which SEMS logs WARN syslog messages. Use 0 to disable warnings. **Changing this value may restart signaling process.** Default value: 5000 API name: redis_warn_threshold
Log every monitored destination state	If enabled, every state of a monitored destination (changed or not) is logged at INFO level. **Changing this value may restart signaling process.** Default value: disabled API name: dest_monit_log_every_state

Parameter Name	Description
Terminate calls on SBC shutdown or restart	If enabled, SBC terminates calls during shutdown or restart. **Info** When using HA, it may be better to keep calls alive across SBC restarts. In such cases, keep this option disabled. Default value: disabled API name: terminate_calls
DNS Resolver Timeout	Timeout (in ms) for DNS query responses. **Changing this value may restart signaling process.** Default value: 100 API name: dns_resolver_timeout
DNS Cache Renew Period	Time (in s) to refresh DNS cache entries before expiry. **Changing this value may restart signaling process.** Default value: 300 API name: dns_cache_early_renew
DNS cache grace period	Time (in s) to keep expired DNS cache entries before removal. **Changing this value may restart signaling process.** Default value: 300 API name: dns_cache_grace_period

Table 13: SEMS Parameters

2.14 SIPREC Parameters

This section contains the configuration parameters related to SIP Recording (SIPREC).

Parameter Name	Description
SIPREC media interface	SBC interface to be used for sending RTP media towards the SIPREC server. **Changing this value may restart signaling process.** Default value: empty API name: siprec_outbound_rtp_interface
SIPREC outbound interface	SBC interface to be used for sending SIP signaling towards the SIPREC server. **Changing this value may restart signaling process.** Default value: empty API name: siprec_outbound_interface
SIPREC SIP timer A (ms)	SIP Timer A used towards the SIPREC server. **Changing this value may restart signaling process.** Default value: 500 ms API name: siprec_timer_A

Parameter Name	Description
SIPREC SIP timer B (ms)	SIP Timer B used towards the SIPREC server. **Changing this value may restart signaling process.** Default value: 32000 ms API name: siprec_timer_B
SIPREC SIP timer C (ms)	SIP Timer C used towards the SIPREC server. **Changing this value may restart signaling process.** Default value: 180000 ms API name: siprec_timer_C
SIPREC SIP timer F (ms)	SIP Timer F used towards the SIPREC server. **Changing this value may restart signaling process.** Default value: 32000 ms API name: siprec_timer_F
SIPREC SIP timer L (ms)	SIP Timer L used towards the SIPREC server. **Changing this value may restart signaling process.** Default value: 32000 ms API name: siprec_timer_L
SIPREC SIP timer M (ms)	SIP Timer M used towards the SIPREC server. **Changing this value may restart signaling process.** Default value: 8000 ms API name: siprec_timer_M
SIPREC SIP timer T2 (ms)	SIP Timer T2 used towards the SIPREC server. **Changing this value may restart signaling process.** Default value: 4000 ms API name: siprec_timer_T2

Table 14: SIPREC Parameters

2.15 SIP Parameters

This panel allows configuring default SIP timer values in the SBC including some extra parameters. For more details, see section SIP Timers.

Parameter Name	Description
SIP timer A (ms)	SIP Timer A used towards SIP peers. **Changing this value may restart signaling process.** Default value: 500 ms API name: sip_timer_A

Parameter Name	Description
SIP timer B (ms)	SIP Timer B used towards SIP peers. **Changing this value may restart signaling process.** Default value: 32000 ms API name: sip_timer_B
SIP timer C (ms)	SIP Timer C used towards SIP peers. **Changing this value may restart signaling process.** Default value: 180000 ms API name: sip_timer_C
SIP timer D (ms)	SIP Timer D used towards SIP peers. **Changing this value may restart signaling process.** Default value: 32000 ms API name: sip_timer_D
SIP timer E (ms)	SIP Timer E used towards SIP peers. **Changing this value may restart signaling process.** Default value: 500 ms API name: sip_timer_E
SIP timer F (ms)	SIP Timer F used towards SIP peers. **Changing this value may restart signaling process.** Default value: 32000 ms API name: sip_timer_F
SIP timer G (ms)	SIP Timer G used towards SIP peers. **Changing this value may restart signaling process.** Default value: 500 ms API name: sip_timer_G
SIP timer H (ms)	SIP Timer H used towards SIP peers. **Changing this value may restart signaling process.** Default value: 32000 ms API name: sip_timer_H
SIP timer I (ms)	SIP Timer I used towards SIP peers. **Changing this value may restart signaling process.** Default value: 5000 ms API name: sip_timer_I
SIP timer J (ms)	SIP Timer J used towards SIP peers. **Changing this value may restart signaling process.** Default value: 32000 ms API name: sip_timer_J

Parameter Name	Description
SIP timer K (ms)	SIP Timer K used towards SIP peers. **Changing this value may restart signaling process.** Default value: 5000 ms API name: sip_timer_K
SIP timer L (ms)	SIP Timer L used towards SIP peers. **Changing this value may restart signaling process.** Default value: 32000 ms API name: sip_timer_L
SIP timer M (ms)	SIP Timer M used towards SIP peers. **Changing this value may restart signaling process.** Default value: 8000 ms API name: sip_timer_M
SIP timer T2 (ms)	SIP Timer T2 used towards SIP peers. **Changing this value may restart signaling process.** Default value: 4000 ms API name: sip_timer_T2
Add Q850 header to timer expiration's CANCEL	If enabled, a Q850 header is added to the CANCEL generated by a timer expiration. Currently, only Timer C is supported. **Changing this value may restart signaling process.** Default value: disabled API name: sip_Q850_expiration
Terminate dialog upon failure replies for in-dialog OPTIONS	Terminates the dialog if in-dialog OPTIONS request fails with reply that should cause dialog termination. Affects only INVITE-based dialogs (i.e., calls). RFC 5057 replies causing termination: 404, 410, 416, 482, 483, 484, 485, 502, 604. Additionally, SBC handles replies 408 and 480 the same way. Purpose: cope with poorly implemented SIP UAs that cannot handle in-dialog OPTIONS correctly. **Changing this value may restart signaling process.** Default value: enabled (terminate the dialog) API name: remote_disappeared_on_options

Parameter Name	Description
Remove filtered m-lines	If enabled, removes media lines filtered out by media whitelist/blacklist. If disabled, they remain in the SDP but are marked as inactive instead. This option is applied globally on all calls with active media whitelist or blacklist (see Media Type Filtering). Purpose: cope with poorly implemented SIP UAs that cannot handle inactive media streams correctly. **Changing this value may restart signaling process.** Default value: disabled (i.e. mark media lines as inactive) API name: remove_filtered_mlines
Try filling missing rtpmap in disabled media	On SDP answers with disabled media, SBC tries to fill in missing a=rtpmap and a=fmtp lines by copying them from the offer if absent. **Changing this value may restart signaling process.** Default value: disabled Available since: 5.3 API name: guess_missing_rtpmap
Filter forced transports	If enabled, removes media lines that do not match outbound transport forced by the Force RTP/SRTP action (see RTP and SRTP Interworking). These lines are left in SDP but converted to the required transport if not enabled. For example: Caller is sending one audio stream over RTP and another audio stream over SRTP (commonly used when SRTP is configured as optional on a phone). SRTP is forced in outbound rules on SBC. If Filter forced transports option is “off” SBC forwards SDP with two audio streams to the callee both of them over SRTP. If this option is “on” SBC forwards SDP with just one audio stream over SRTP to the callee. This option is applied globally on all calls using Force RTP/SRTP action. Purpose: cope with interoperability issues of SIP UAs that cannot handle multiple media streams of the same type. **Changing this value may restart signaling process.** Default value: disabled (i.e. convert the media lines to the forced transport) API name: filter_forced_transports

Parameter Name	Description
Call transfers using late offer-answer	If enabled, uses offer-less INVITE when generating a new call leg during call transfer (unattended call transfer or call transfer replacing non-local call). Reliable, but many SIP UAs fail to implement late offer-answer correctly. **Changing this value may restart signaling process.** Default value: disabled API name: xfer_using_late_oa
Predefined payloads for call transfers	Comma-separated list of codecs to add into SDP of INVITE generated during call transfer (unattended call transfer or call transfer replacing non-local call). If empty, only codecs used in the call are included, which may cause issues if the peer does not support them. Only simple codecs can be used (no parameters can be specified). Example: PCMU,PCMA **Changing this value may restart signaling process.** Default value: empty API name: xfer_static_payloads
Force outbound interface	If enabled, UDP packets are forced to use the system interface attached to the outbound call agent. **Info** This option relies on operating system capabilities that have heavy limitations. Especially, when forcing the outbound interface, the Linux IP stack will set the source IP on its own, which might lead to unwanted effects (invalid source IP that e.g. SEMS might not be using at all). In many cases, this option will not effect the desired functionality and is not recommended. Manually configured source IP based policy routing is the preferred method. **Changing this value may restart signaling process.** Default value: disabled API name: force_outbound_if

Table 15: SIP Parameters

2.16 SRTP Parameters

These parameters define the security handshake for Secure RTP (SRTP). SRTP is always used for WebRTC and may also be used with certain SIP devices that support encryption.

Parameter Name	Description
DTLS certificate file	Certificate file. Optional. Leave empty for self-signed certificate. This is the recommended configuration, since other certificates may cause DTLS packets to become too large and fail to traverse NATs due to IP fragmentation. **Changing this value may restart signaling process.** Default value: empty API: dtls_certificate
DTLS private key file	Private key file. Optional. **Changing this value may restart signaling process.** Default value: empty API: dtls_key
DTLS CA list file	CA list file. Optional. Leave empty if don't want to verify the remote DTLS certificates with a CA. **Changing this value may restart signaling process.** Default value: empty API: dtls_ca_list
DTLS handshake timeout (ms)	Duration in milliseconds for the DTLS handshake to complete before terminating the call. Use 0 to disable. **Changing this value may restart signaling process.** Default value: 5000 API: dtls_timeout
SRTP crypto-suite AES_CM_128_HMAC_SHA1_32	Enables or disables the crypto suite AES_CM_128_HMAC_SHA1_32. It should remain enabled unless required otherwise for interoperability. **Changing this value may restart signaling process.** Default value: enabled API: srtp_support_hmac_sha1_32_suite
SRTP crypto-suite AES_CM_128_HMAC_SHA1_80	Enables or disables the crypto suite AES_CM_128_HMAC_SHA1_80. It should remain enabled unless required otherwise for interoperability. **Changing this value may restart signaling process.** Default value: enabled API: srtp_support_hmac_sha1_80_suite
SRTP crypto-suite AES_256_CM_HMAC_SHA1_80 (SDES only)	Enables or disables the crypto suite AES_256_CM_HMAC_SHA1_80 (SDES only). It should remain enabled unless required otherwise for interoperability. **Changing this value may restart signaling process.** Default value: enabled API: srtp_support_aes_cm_256_suite

Parameter Name	Description
SRTP crypto-suite AEAD_AES_128_GCM	Enables or disables the crypto suite AEAD_AES_128_GCM. It should remain enabled unless required otherwise for interoperability. **Changing this value may restart signaling process.** Default value: enabled Available since: 5.2 API: srtp_support_aead_aes_128_gcm_suite
SRTP crypto-suite AEAD_AES_128_GCM_8 (SDES only)	Enables or disables the crypto suite AEAD_AES_128_GCM_8 (SDES only). It should remain enabled unless required otherwise for interoperability. **Changing this value may restart signaling process.** Default value: enabled Available since: 5.2 API: srtp_support_aead_aes_128_gcm_8_suite
SRTP crypto-suite AEAD_AES_256_GCM	Enables or disables the crypto suite AEAD_AES_256_GCM. It should remain enabled unless required otherwise for interoperability. **Changing this value may restart signaling process.** Default value: enabled Available since: 5.2 API: srtp_support_aead_aes_256_gcm_suite
SRTP crypto-suite AEAD_AES_256_GCM_8 (SDES only)	Enables or disables the crypto suite AEAD_AES_256_GCM_8 (SDES only). It should remain enabled unless required otherwise for interoperability. **Changing this value may restart signaling process.** Default value: enabled Available since: 5.2 API: srtp_support_aead_aes_256_gcm_8_suite
SRTP crypto-suite preference order	Comma-separated list of crypto suites, e.g., 'AEAD_AES_256_GCM, AEAD_AES_128_GCM'. Suites offered by the remote endpoint take precedence. Suites supported but not listed here are appended at the end according to default order. **Changing this value may restart signaling process.** Default value: empty Available since: 5.2 API: srtp_support_suite_order

Parameter Name	Description
MTU to use with DTLS (ms)	MTU value (in milliseconds) to use with DTLS. Use 0 for the interface default. **Changing this value may restart signaling process.** Default value: 0 API: dtls_config_mtu
DTLS fallback MTU (ms)	MTU value (in milliseconds) used only if the normal MTU is too large and causes socket errors. **Changing this value may restart signaling process.** Default value: 1200 API: dtls_config_fallback_mtu
DTLS legacy server connect	Enables or disables support for connections to legacy DTLS servers that do not support secure renegotiation extension. By default, the DTLS handshake is terminated if this is not enabled. See RFC5746 sections 3.4 and 4.1 for details. WARNING: Enabling this reduces security and should only be used when the DTLS servers the SBC connects to have renegotiation disabled. **Changing this value may restart signaling process.** Default value: disabled Available since: 5.4 API: dtls_legacy_server_connect

Table 16: SRTP Parameters

2.17 Signaling SSL

This section contains the configuration parameters related to signaling SSL.

Parameter Name	Description
Revoked certificates (CRL) file	CRL file holding a list of revoked certificates. Used only by the SEMS signaling process. **Changing this value may restart signaling process.** Default value: empty API: ssl_crl_file

Parameter Name	Description
Minimal supported TLS version	The minimal supported TLS version on signaling interfaces. Supported values: tls1, tls1.1, tls1.2, tls1.3. **Changing this value may restart signaling process.** Default value: tls1.2 API: tls_min_version Possible values over API: • tls1 • tls1.1 • tls1.2 • tls1.3
TLS cipher list	The supported TLS cipher list for signaling interfaces. OpenSSL syntax is used. **Changing this value may restart signaling process.** Default value: HIGH:!aNULL:!MD5:!DH:!SHA1:!SHA256:!SHA384:!RSA API: tls_cipher_list
TLS EC curves list	Colon-separated list of EC curves used with TLS for signaling interfaces. Each item is a curve NID or name, for example "P-521:P-384:P-256". **Changing this value may restart signaling process.** Default value: empty API: tls_ec_curves_list
Dump TLS session keys to file	If enabled, TLS session keys will be dumped to a file for diagnostics (/data/pcap/tls_keys). Requirements: This option must be enabled if downloading a diagnostic bundle from the GUI that includes both pcap files and TLS keys. Otherwise, only pcap files are included. Limitations: WebRTC interface is not supported. **Changing this value may restart signaling process.** Default value: disabled API: dump_tls_keys

Table 17: Signaling SSL Parameters

2.18 Syslog Parameters

These parameters allow fine-tuning of the syslog daemon behavior. They are primarily useful when syslog messages are configured to be sent to an external system.

Parameter Name	Description
Log level	Sets the global SEMS syslog log level. See section Reference of Log Level Parameters for the full list of options. **Changing this value may restart signaling process.** Default value: 2 API: sems_log_level

Parameter Name	Description
Syslog facility	Syslog facility to use for logs generated by the main SBC processes. Possible values: 'daemon', 'user', 'local0' … 'local7'. Default value: daemon API: syslog_facility Possible values over API: • daemon • user • local0 • local1 • local2 • local3 • local4 • local5 • local6 • local7
Enable remote syslog servers	If enabled, syslog messages will also be sent to an external syslog host in addition to being written to the local file- system. Default value: disabled API: syslog_remote_enabled
Remote syslog server address	IP address of the external syslog server. Default value: 127.0.0.1 API: syslog_remote_ip
Remote syslog server port	Port number on which the external syslog server listens. Default value: 514 API: syslog_remote_port
Remote syslog transport	Transport protocol for the external syslog server. Supported values: 'udp', 'tcp' or 'tls'. Default value: udp API: syslog_remote_transport Possible values over API: • udp • tcp • tls

Parameter Name	Description
Log level for remote syslog server	Minimum severity of messages sent to the external syslog server. Supported values: 'emergency', 'alert', 'critical', 'error', 'warning', 'notice', 'info', 'debug'. Default value: error API: syslog_remote_loglevel Possible values over API: • emergency • alert • critical • error • warning • notice • info • debug
Log files rotation frequency	Frequency of rotating local log files. Supported values: "daily", "weekly", or "monthly". Default value: daily API: syslog_logrotate_freq Possible values over API: • daily • weekly • monthly
Number of old log files to keep	Number of rotated log files to retain before deletion. Default value: 7 API: syslog_logrotate_times
Secondary remote syslog server address	IP address of a secondary external syslog server. Use empty to disable. Default value: empty API: syslog_remote_ip2
Secondary remote syslog server port	Port number of the secondary external syslog server. Default value: 514 API: syslog_remote_port2
Secondary remote syslog transport	Transport protocol for the secondary external syslog server. Supported values: 'udp', 'tcp' or 'tls'. Default value: udp API: syslog_remote_transport2 Possible values over API: • udp • tcp • tls

Parameter Name	Description
Log level for secondary remote syslog server	Minimum severity of messages sent to the secondary external syslog server. Supported values: ‘emergency’, ‘alert’, ‘critical’, ‘error’, ‘warning’, ‘notice’, ‘info’, ‘debug’. Default value: error API: syslog_remote_loglevel2 Possible values over API: • emergency • alert • critical • error • warning • notice • info • debug
Send CDRs to remote syslog server	If enabled, Call Detail Records (CDRs) will be included in log messages sent to the remote syslog server. Default value: disabled API: syslog_remote_cdr_enabled Possible values over API: • daily • weekly • monthly

Table 18: Syslog Parameters

2.19 System Monitoring Parameters

These parameters configure email alerts that are triggered when system resources are excessively used. The same email address is also used for setting up automatic Let’s Encrypt certificate renewal.

Parameter Name	Description
Email address for sending alerts	Email address to which important alerts such as excessive CPU usage reports are sent. Leave empty to disable email alerts. This address is also used for Let’s Encrypt TLS certificate auto-renewal. Default value: empty API: alert_email
From email address for sending alerts	Available since: 4.3 Email address used as the “From” field in alert emails. If empty, the system default is used. Default value: empty API: alert_emailfrom
SMTP mail server address for sending alerts	Address of the SMTP server used to send alert emails. When SBC runs in a container, a local mail relay is unavailable, so an external mail server must be configured. Default value: empty API: alert_mailserver

Parameter Name	Description
SMTP mail server port	Port number of the SMTP mail server. Default value: 25 API: alert_mailserver_port
Use secure connection to SMTP mailserver	Defines whether the SMTP connection should be encrypted, and if so, whether TLS or STARTTLS is used. Default value: no API: alert_mailserver_tls Possible values over API: • no • TLS • STARTTLS
SMTP mail server authentication	Enables or disables SMTP authentication. If set to “on”, the authentication type is chosen automatically. Default value: off API: alert_mailserver_auth Possible values over API: • off • on
Username for SMTP authentication	Username used for SMTP authentication, if authentication is enabled. Default value: empty API: alert_mailserver_user
Password for SMTP authentication	Password used for SMTP authentication, if authentication is enabled. Default value: empty API: alert_mailserver_pass
Send system monitoring data to ABC Monitor	If enabled, system monitoring data is sent to the remote ABC Monitor together with signaling events. Default value: enabled API: collectd_to_console
Send extended system info emails when threshold is exceeded	If enabled, detailed system information is included in alert emails when monitoring thresholds are exceeded. Default value: enabled API: alert_email_info
Extended info emails frequency	Defines the minimum interval between sending extended system info emails. The value must include a suffix: “min”, “hour”, or “day” (e.g., 10min, 1hour, 1day). Default value: 1hour API: alert_freq

Parameter Name	Description
Check status of system interfaces	If enabled, system network interfaces are periodically checked and alert events are generated if errors are detected. Specific check types can be configured using related parameters. Default value: enabled API: checknet_interfaces_chk
Include only MI/SI interfaces in status check	Defines whether the system network interface checks include only media and signaling interfaces or all interfaces. Default value: enabled API: checknet_interfaces_misi
Interval for the system interfaces checks	Interval in seconds at which system network interfaces are checked. Default value: 300 API: checknet_wait

Table 19: System Monitoring Parameters

2.20 Probe Parameters

This section contains the configuration parameters related to Probe operation and management.

Parameter Name	Description
Log level	General log level for probe logging. Default value: info API: probe_log_level Possible values over API: • "-3": bug • "-2": critical • "-1": error • "0" : warning • "1" : notice • "2" : info • "3" : debug
Log message options	Log message options (bitmask of flags): • 1: short filename:line • 2: long format • 4: timestamp • 8: backtrace short • 16: backtrace long Default value: 1 API: probe_log_opt

Parameter Name	Description
Parse / capture log level	Log level for parsing/capturing SIP messages. Default value: notice API: probe_parse_log_level Possible values over API: • "-3": bug • "-2": critical • "-1": error • "0" : warning • "1" : notice • "2" : info • "3" : debug
Parse / capture log message options	Log message options for parsing/capturing SIP messages (bitmask of flags): • 1: short filename:line • 2: long format • 4: timestamp • 8: backtrace short • 16: backtrace long Default value: 4 API: probe_parse_log_opt
Extra REGISTER expiration delta	Extra REGISTER expiration delta (in seconds) used to absorb delayed re-REGISTERs. Default value: 30 API: probe_reg_exp_delta
Ignore port number when comparing contacts	If enabled, the port number is ignored when comparing contacts (but not AORs). Default value: disabled API: probe_contact_ignore_port
Interval for generating statistics/counters events	Interval in seconds for generating statistics and counters events. Default value: 300 API: probe_stats_interval
Event rate max values list	List of event rate maximum values. If any is exceeded, the event is blacklisted. Use 0 to disable a rate. Example for 2 rates: 20,240,0. Default value: 16,240,600 API: probe_evr_limits
Event rate time intervals	Time intervals corresponding to event rate limits, used to calculate the rates. Use 0 to disable. Default value: 1s,1m,10m API: probe_evr_intervals

Parameter Name	Description
Event rate exponential back-off min	Report blacklisted events only if the number of "repetitions" is a multiple of this value and 2^k and n < Event rate exponential back-off max. To disable exponential back-off for the blacklist re-reporting, set the value to 0 or to Event rate exponential back-off max. Default value: 100 API: probe_evr_conseq_report_min
Event rate exponential back-off max	Maximum repetition count for re-reporting blacklisted events. Use 0 to disable. Default value: 10000 API: probe_evr_conseq_report_max

Table 20: Probe Parameters

Chapter 3

Reference of Log Level Parameters

In several ABC SBC configuration places, the log reporting levels may be configured. The ABC SBC allows to set the logging levels both globally and by functional areas. The increase log level may help with troubleshooting however caution is advised. Increased log level can dramatically degrade system performance.

This reference provides explanation how to set the proper logging level. Log levels are represented with an integer value and have the following possible values:

- 0 / ERROR
- 1 / WARNING
- 2 / INFO
- 3 / DEBUG

If only log-level is set, it is used globally. The log level can be changed however for only some specific functional area by preceding the value with "Category:Subcategory=" expression. Multiple such expressions can be combined with each other using semicolon as shown in the following example:

```
1;SIP:Transaction=3;SDP:Parser=3;RTP:*=3;PLUGIN:sbc=3
```

This example sets the default log level to 1, whereas SIP transaction machine, SDP parser, RTP engine and SBC logic reports at log level 3.

Category	Subcategory
Core	• Main • Config • Thread • Timer • Events • SessionContainer • SessionProcessor • SessionWatcher • MediaProcessor • Plugin • Utils
SIP	• Ctrl • Parser • Transport • Transaction • Dialog • OfferAnswer • Session • Registration • Subscription • DNS • Blacklist
B2B	• B2BSession • B2BMedia

SDP	• Parser • MimeBody
RTP	• Stun • RtpPacket • RtcpPacket • RtpTransport • RtpStream • RtpAudio • DTMF
SRTP	• SRTP • SDES • DTLS • ZRTP • Socket
AUDIO	• Audio • AudioFile • AudioMixer • Conference • Playlist • Prompt • Jitter
PLUGIN	• all • dsm • sbc • redis_store • websock • reg_agent • cc_gui • cc_gui_rules • cc_gui_xfer • cc_gui_dest • sbc_replication • rest • webconference • xmlrpc2di

Table 21: Log Level categories

Chapter 4

Reference of Call Agent Configuration Parameters

This reference lists all Call Agent configuration parameters used in ABC SBC. These parameters take effect on any traffic that is specific to a Call Agent without need to place any additional action into the Call Agent's rulebase.

The actions are grouped as follows:

4.1 Destination Monitor Parameters

These parameters configure health checks on Call Agents by sending OPTIONS requests at regular intervals.

Depending on whether the Call Agent responds to these OPTIONS requests, its destinations can be added to a destination blacklist, thereby removing them from the pool of potential target destinations.

If blacklisting is enabled, it is also possible to configure a list of SIP reply codes that, if received, will also mark the destination as unavailable.

Parameter Name	Description
Monitoring interval (sec)	Interval between sending OPTIONS-based health-checks to the monitored Call Agent. If zero, no monitoring takes place.
Max-Forwards	Value of Max-Forwards header field in the health checking OPTIONS requests.
Blacklist TTL (seconds)	The period of time an unresponsive address remains on the blacklist. If zero, blacklisting is not used.
Unavailable on Reply Codes	Comma separated list of SIP Response codes

4.2 Failover Parameters

These parameters allow to define when a new destination is tried. By default, this occurs if a destination fails to respond within a predetermined timeframe. However, it is possible to configure a list of SIP Response codes that will produce the same effect, triggering a failover to the next available destination.

It is also possible to add destinations that have been found to be unresponsive (either through a timer or due to a specific SIP reply code) to s destination blacklist.

See the Section IP Blacklisting: Adaptive Availability Management for additional information.

Parameter Name	Description
On Reply Codes	Comma separated list of SIP Response codes
Blacklist TTL (seconds)	The period of time an unresponsive address remains on the blacklist. If zero, blacklisting is not used.
Blacklist grace timer (milliseconds)	Additional period of time to provide a safety buffer in case that conflicting timers occur along a SIP path.

4.3 Registration Agent Parameters

Registration agent allows to register the ABC SBC with a third-party SIP service be sending pre-defined REGISTER requests as described in the Section Registration Agent. The following Call Agent parameters define if such a registration agent shall be active and how its registration parameters shall be formed.

Registration agent credentials set on the call agent will also be used for authorization requests for calls as well, unless overridden by auth actions.

Parameter Name	Description
Enabled	Turns a registration agent on or off.
URI domain.	Domain name to be used in REGISTER requests URIs
URI name.	User name to be used in REGISTER request URIs
Display name	Display names as included in the From header-field of the REGISTER requests
auth name	SIP User id as used in the authentication header fields. May be different from user names in URIs.
auth password	SIP user password used in the digest authentication
Contact	Content of the Contact header-field in the REGISTER requests. Specific usernames may be chosen to make it easier to identify incoming requests coming to addresses registered using the registration agent.
Contact HF Params	Semi-colon separated header parameters to add to the Contact header.
Additional headers	\r\n-separated headers to add to the requests. I.e. 'x-my-hdr: v1\r\nx-my-hdr2: v2'.
Registration interval (seconds)	Time between subsequent registrations are sent
Retry interval (seconds)	Period of time to keep till the next attempt when the previous failed
Next Hop (IP address)	Address of a destination to which a request will be sent
Registrar affinity	Binding of the registrar. `Sticky` mode records the reply IP/Port/Transport and initially tries that for refreshing the registration. `Lazy` is same as sticky except that it does a lookup of the recorded reply IP address in the SBC's internal reverse-dns cache table and discards the record if it is not found in the cache. `Active` does not record the reply address at all. Limitations: • In `Lazy` mode, only the IP address is checked for existance in the cache and not port & transport. • Items in the reverse-dns cache are still considered valid after their expiry, until the duration specified in the `DNS Cache Grace Period` global configuration passes. Available since: 5.2

Bulk Contact	Turn on to support the SIP bulk contact registration form as described in RFC3680.

Table 22: Registration Agent Parameters

4.4 Topology Hiding Parameters

The Section Topology Hiding discussed purpose and use of Topology Hiding. The following options enable/disable this functionality for the respective Call Agents.

Parameter Name	Description
Enabled	Turning this option replaces occurrences of IP addresses in well-known header-fields of SIP signaling with those of the ABC SBC .
Cross-Realm	If enabled, topology hiding is used even when signaling ingress and egress realms are the same.

4.5 Firewall Blacklisting Parameters

Automated IP address blocking is discussed in the Section Automatic IP Address Blocking. Several attributes defined what kind of Call Agent behavior adds to the score that may eventually lead to blacklisting of the source IP address.

Parameter Name	Description
Sanity	If turned on, invalid SIP messages add to the auto-blocking score and may lead to blocking of their originator. Otherwise they are silently ignored.
Auth	If enabled, failed authentication add to the auto-blocking score and may lead to blocking of their originator. Otherwise only events are reported but no further action is taken.

4.6 Security Parameters

Parameter Name	Description
Don't expect authentication	Don't expect any authentication on this call agent. Drops any 401/407 replies from this agent. Removes 'Authorization' and 'WWW-Authorization' sent towards this agent, 'Proxy-Authenticate' and 'WWW-Authenticate' headers received from this agent.

4.7 SIP Timer Parameters

Parameter Name	Description
SIP Timer [X]	Allows setting SIP timers per agent. Each SIP timer set overrides the global configuration.
Failover reduce factor	`Failover reduce factor` is used to divide B, F & M timers when the destination call agent has a backup CA. This allows for a faster failover. Leaving it empty uses the default value of 4.

Please refer to section SIP Timers for more details.

4.8 Resolver Parameters

Parameter Name	Description
Nameserver IP addresses (comma-separated)	DNS nameservers to use while communicating through this call-agent. Each unique nameserver configuration has its own reverse-dns-cache. If parameters of two configurations are the same (i.e. regardless of the order, same set of nameservers & bind-to-ip address flag resolves to the same physical interface) then they share a common reverse-dns cache. This rule covers the DNS configuration in the signaling interfaces as well. If this is set, it will get used as soon as this call agent is chosen. Until the CA is chosen, either the signaling interface's configuration will be effective, or if that does not exist, system's configured nameservers will be used. When trying to find a source call-agent that is identified by DNS, a DNS reverse-cache lookup is done using the source IP. This look-up follows these steps until a match is found: 1. A reverse-cache search is done on the resolver of each call-agent that is assigned to the signaling interface that the SIP message came from. If such a call-agent does not have a nameserver configuration, then the look-up is done on the system-level resolver for that call-agent. 2. A reverse-cache search is done on the resolver of the signaling interface. If the signaling interface does not have a nameserver configuration, then the look-up is done on the system-level resolver. The same look-up logic applies to finding a destination call-agent as well. This configuration is per-leg. Registration Agent also makes use of this configuration.
Bind to signaling interface	Strictly use the underlying physical interface of the signaling interface of this call-agent.

Chapter 5

Default Audio Files

Most of the prompts' sample rate is 8000. It isn't necessarily required, as `sems` resample them. Note that wideband samples may sounds nicer.

All of the meet-me actions' offer two sets of defaults audio prompts:

- */usr/lib/sems/audio/webconference* (English)
- */usr/lib/sems/audio/webconference/de* (German)

Multi-lingual support can be used in conjuncture with those 2 directories. See Multi lingual conferencing announcements for more information about that feature.

5.1 Join meet-me conference

The following prompts are used by multiple meet-me conference configuration.

Audio file	Content
General audio files	
contact_support	Please contact support.
enter_pin	Please enter your code, then press the pound key.
entering_conference	You are now entering your conference room.
first_participant	**ton** Welcome, you are the first participant in the conference.
max_attempt_reached	We are sorry you are having problems. Please try later or contact customer support.
please_enter_room	Please enter your conference room, then press the pound key.
please_enter_your_code	Please enter your code, then press the pound key.
short_pin	This PIN is too short. Please try again.
simple_pin	This PIN is too simple. Please try again.
room_created	Room created.
timeout_enter_pin	This input unfortunately took to long. Please try again later.
yourcodeis	Your code is
yourroomnumberis	Your room number is
welcome	Welcome. This is FRAFOS conference.
wrong_pin	This code is not correct. Please try again.
wrong_pin_bye	This code is not correct. Please try again later or contact customer support.
x_welcome_and_prompt	Welcome this is FRAFOS' conference. Please enter your code, then press the pound key.
join_sound / drop_sound	**biip** / **buup**
Security PIN audio files	
andpinis	And the PIN is
create_secu_pin	Please enter a PIN for the new room, followed by the pound key.

Audio file	Content
enter_secu_pin	Please enter the PIN of the room, followed by then pound key.
repeat_secu_pin	Please repeat the new PIN, followed by the pound key.
secu_pin_set_to	PIN set to
secu_pin_3_digits	Sorry, security PIN must be at least 3 digits.
Record username audio files	
current_participants_are	**The current participants in the conference** are…
just_joined_conf	… just joined the conference
just_leaved_conf	… just leaved the conference
recording_1_2_3	To keep this recording, please press 1, To replay the recording, please press 2. To record your name again, please press 3.
say_ur_name	Please, say your name after the tone. Then, press the pound key.
timeout_record	Username recording timed out. Please try again later.
ur_name_is	Your recorded name is
Generate room audio files	
ask_if_gen	To enter a conference room, please press 1. To create a new room, please press 2
error_persist_room	An error occurred while saving the new room and PIN.
generating_room	We are now creating a conference room
repeat_or_enter	Press 1 to hear room number an pin again. Press 2 to go into your room.
timeout_generate_room	This input unfortunately took to long. Please try again later.
Multi lingual support audio files	
select_lang	To continue in English, press one. Um auf Deutsch vor zu fahren, drücken Sie bitten bis zwei

Please note that digits prompts are also needed. When multi-lingual isn't used, files are expected to be found in the same directory as the matching Conferencing' global config. In case of multi-lingual, files are expected to be found in the *digits/* sub-directory.

SBC support two kind of number echoing:
- left to right: Forty two
- right to left: Zwei Und Vierzig

LtR expected files are the following:
- digits: 0.wav, 1.wav, 2.wav, 3.wav, 4.wav, 5.wav, 6.wav, 7.wav, 8.wav, 9.wav
- multiple of 10: 10.wav, 20.wav, 30.wav, 40.wav, 50.wav, 60.wav, 70.wav, 80.wav, 90.wav
- tens: 11.wav, 12.wav, 13.wav, 14.wav, 15.wav, 16.wav, 17.wav, 18.wav, 19.wav
- 21 to 99: x2.wav, x3.wav, x4.wav, x5.wav, x6.wav, x7.wav, x8.wav x9.wav

RtL expected files are the following:
- digits: 0.wav, 1.wav, 2.wav, 3.wav, 4.wav, 5.wav, 6.wav, 7.wav, 8.wav, 9.wav
- multiple of 10: 10.wav, 20.wav, 30.wav, 40.wav, 50.wav, 60.wav, 70.wav, 80.wav, 90.wav
- tens: 11.wav, 12.wav, 13.wav, 14.wav, 15.wav, 16.wav, 17.wav, 18.wav, 19.wav
- 21 to 99: 2x.wav, 3x.wav, 4x.wav, 5x.wav, 6x.wav, 7x.wav, 8x.wav 9x.wav

5.2 Meet-me set PIN audio prompts

Audio file	Context	Content
setPin_welcome	Use for welcome	'welcome, you can set a pin for your personal conference room with the number' ...
setPin_welcome_set	Used to welcome when the security PIN is already set.	'welcome, your personal conference room with the number' ...
setPin_enter_pin	Used to prompt the security PIN	'please enter the security pin of the room number' ...
setPin_change_pin	Used to prompt the security PIN	'please hang up if you want to keep it, otherwise'
setPin_repeat_pin	Used to confirm the security PIN user	'please repeat the pin and and press the pound key'
setPin_pin_set	Used in case of success	'your pin was successfully set, thanks you.'
setPin_pin_dont _match	Used when user PIN don't match	'the pin numbers you've enter does not match. Please try again, and enter a new PIN, followed by the pound key.'
setPin_failed	Used in case of failure	'please hang up if you want to keep it, otherwise'

Table 23: Audio prompts

5.3 Two-Factor authentication

Audio file	Context	Content
2fa_greeting	Use for welcome	'Please enter the two factor authentication PIN number that was set for this line
2fa_pin_correct	Used in case of success	'that is correct, thanks you. Please hold the line to be connected'
2fa_failed	Used to prompt the security PIN	'I'm sorry you're having entering the pin number. Please hold the line to be connected to the help desk.'
2fa_pin_wrong	Used to prompt the security PIN	'sorry this is not correct. Please enter the 2 factor authentication pin number that we set for this line.'

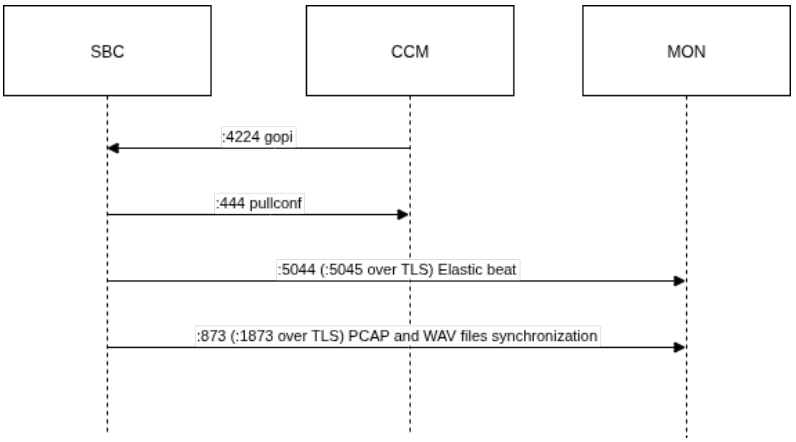
Table 24: Audio prompts

Chapter 6

Reference of Default Port Numbers

The reference lists port numbers the ABC SBC, Cluster Config Manager and ABC Monitor uses. It is particularly useful when considering firewall policies for firewalls placed in front of the ABC SBC. The reference lists default port numbers, transport protocols, container opening the port, service listening on that port and the interface on which the respective applications are permitted. In addition to the SBC interfaces (see SBC Interfaces), some applications may be listening on all interfaces while some management applications are using the loopback interface for internal communication.

Note that while the ABC SBC only accepts traffic on the ports and interfaces specified in the following specification, further restrictions may apply. Signaling is only accepted from well-defined Call Agents and certain traffic may be blacklisted (see Manual SIP Traffic Blocking).



Port	Container	Description
22	ABC SBC	(ssh / TCP) Secure shell server. Used for remote management. Value 0 can be used for default port, which is 24. It can be set using ssh app on SBC interface.
25	ABC SBC	(SMTP / TCP) Local Email relay. Used to forward email alerts. From outside perspective it acts as a client.
161	ABC SBC	(SNMP / UDP) Internal SNMP management.
443	Cluster Config Manager	(HTTPS / TCP) Administrative GUI.
444	Cluster Config Manager	(HTTPS / TCP) Allow ABC SBC to download new configuration and upload status file to the Cluster Config Manager.
1443	ABC SBC	(HTTPS / TCP) XML-RPC provisioning.
3306	Cluster Config Manager	(TCP) MySQL database.
5060	ABC SBC	(sip / UDP, TCP) SIP signaling.
5061	ABC SBC	(sip / TLS)SIP signaling over TLS.
6379	ABC SBC	(TCP) redis replication, if HA is used.
8080, 8081	ABC SBC	(TCP) SIP over Websocket WebRTC.
8090	ABC SBC	(TCP) XML-RPC remote programming interface

10000 to 60000	ABC SBC	(UDP) Audio/video media.
15441, 4443		(TCP) webconference demo. available only on request.
1444	ABC SBC	(TCP) RESTful port for AWS SNS, disabled by default.
4247, 4248	ABC SBC	(TCP) sbc-eventbeat-[1,2] Expose live metrics and statistics about the redis queues event processing. Local use on localhost, SBC node only.
4224	ABC SBC	(HTTPS) sbc-gopi unified RESTful json API, allowing various interactions with the ABC SBC node. By default the 4224 is protected by firewall and accessible only from CCM. However if IMI interface is configured with a valid TLS certificate and *Verify peer certificate* is enabled, the firewall rule is removed and port protection relies on TLS verification only.

Additional fixed source port numbers shall be opened for the ABC SBC acting as client reaching outside servers as listed in the following table:

SBC Client Port	**Description**
NTP/123/UDP	Time Synchronization
domain/53/UDP	DNS Resolver

Other applications running on the ABC SBC use external applications while locally binding to ephemeral ports.

Remote Server Port	**Description**
syslog/514	remote syslog facility if configured under Global Config / syslog-ng
rsync/873	remote PCAP/WAV storage if enabled under Global Config / replicate recordings / traffic log
rsync/1873	remote PCAP/WAV storage if enabled under Global Config / replicate recordings / traffic log, using TLS if secure connection to ABC Monitor enabled
6379,redis	redis replication and event generation to a ABC Monitor
16379,redis	redis replication and event generation to a ABC Monitor over TLS if enabled

Chapter 7

Reference Interface Parameters

The following parameters can be defined at interface level:

Parameter Name	Description
force_via_address	When enabled, incoming requests are replied to the address shown in their Via header field. This conforms to the RFC3261 specification but often fails to traverse NATs and also permits a reflection attack through the ABC SBC.
no_rport	When enabled, the *rport* parameter is not added to the first *Via* header field.
wspath_xxx	The option, where xxx can be set as needed, sets up an HTTP proxy from path /xxx on HTTPS 443 port (or other port number if using a non-standard one) to the Websocket port on localhost . (It has to be used only on interface using system interface “lo”.

Chapter 8

Reference Application Interface Options

Starting 4.5, the ABC SBC offers the possibility to configure some application option per logical interface, allowing a better control over which process is listening on which port.

Some applications require a TLS profile assigned to corresponding SBC or applied interface.

From SBC release 5.2 and up to 5.4, the following applications were available:

- SSH
- Media
- Signaling
- WebSocket signaling
- SNMP
- Prometheus Pull Service
- TURN server for websocket
- Local monitoring query service
- PCAP query service
- HA vrrp and call state replication
- Local webconf API
- Management for host
- HTTP proxy
- HTTP redirect
- frafos-logprovider
- Log files provider (`frafos-logprovider` has been replaced by it in 5.1).
- Local packet classifier

Starting 5.5, the following applications have been made available, but not configurable:

- Unified SBC management service

It act as a replacement for:

- Local monitoring query service
- PCAP query service
- Local webconf API
- Management for host
- Log files provider
- Local packet classifier
- Prometheus Pull Service

As such, those applications are deprecated, marked as `For nodes up to version 5.4 only` in the UI. If added to an interface, they'll be highlighted in beige. Please note that this UI behavior can be tweaked depending on the selected value for the CCM compatibility mode.

Following the migration from a systemd based environment to the new-generation s6 one, the following applications were dropped:

- SSH
- SNMP
- TURN server for websocket
- HTTP proxy
- HTTP redirect

The following key words in this document are to be interpreted as:
- `IMI`: internal management interface
- `SI`: signaling interface
- `MI`: media interface
- `WS`: websocket interface
- `CX`: custom interface

See Legacy application for legacy applications.

8.1 Supported by the current release

8.1.1 Unified SBC management service

> **Info**
>
> Application available since SBC release 5.5.

The SBC node now ships a single RESTful API listening on the port **4224**: `gopi`.

This API unifies the functionality of the legacy APIs, aka: Local monitoring query service, PCAP query service, Local webconf API, Management for host, Log files provider, Local packet classifier and `goconf`.

As such, the API allows to:
- fetch metrics from various sources (redis list, SEMS's xmlrpc API)
- generate and serve PCAP files based on an aggregation of the one available on the SBC node file system
- expose information and actions related to SEMS's web-conferencing features
- run various administrator tasks from RESTful endpoints
- access the node's file system log files
- partial interaction with the node's firewall
- publish new configuration and provisioned tables to the node
- fetch the node status

The API will by default listen on any IP (`0.0.0.0`) over HTTPS, allowing the Cluster Config Manager to push the initial configuration to it. The default listening IP address can be changed by setting the environment variable `SERVER_IP` to the desired IP.

When the ABC SBC container is started in push mode (`CONFIG_MODE=push`), it normally still performs an initial configuration pull from the Cluster Config Manager at startup, using `MASTER` as the Cluster Config Manager address and `CONFIG_USER` / `CONFIG_PASS` for HTTP basic authentication on the pullconf API. Setting the environment variable `NO_BOOTSTRAP_PULL` to `1` skips this initial pull, so that the node simply waits for the Cluster Config Manager to push the first configuration.

Pushing a configuration to the ABC SBC API requires a valid API token. To avoid a chicken-and-egg situation on a freshly started node that has not yet received any configuration, `NO_BOOTSTRAP_PULL=1` additionally causes token authorization on the ABC SBC API to be disabled until a configuration containing tokens is activated on the node.

See Configuring the SBC container for a complete description of the environment variables that control how the ABC SBC container connects to the Cluster Config Manager at startup.

For every valid configuration published to the SBC node, the process will (re-)start listening on any `IMI` interface, using the attached TLS profile. The API's swagger documentation can be accessed from any browser where the SBC node's IP can be reached, at `https://[SBC IMI IP]:4224/`.

Please note that the API is **not** configurable, albeit one may enable/disable it from an `IMI` interface, allowing the reduce the bloated UI. One may also hide the application from the CCM UI screen by setting the compatibility mode to something **lower than** `5.5` (ex: `5.4`). In any ways, please note that having the application enabled on a 5.4 or earlier node **won't** affect that particular node.

8.1.2 Media

The media application impacts SBC communication handling. Note that this application only has effect on SBC node.

The application is exclusive and mandatory to `MI` interface.

The port range specifies a UDP port range used for media traffic, and does not use TLS.

Parameter Name	Description
Ports	Port range on which SBC may open a socket for media communications.
TOS	This sets "type of service" field in IP packets header. Default value: 184

8.1.3 Signaling

The signaling application impacts SBC communication handling. Note that this application only has effect on SBC node.

If "TLS Port" is not empty, a TLS profile is required.

The application is exclusive and mandatory to `SI` interface.

Parameter Name	Description
Port	Ports on which SBC will open a signaling socket.
TLS Port	(optional) TLS port on which SBC opens a socket for secured signaling communication. SEMS will not use TLS for signaling, if this value is left empty.
Interface Options	Special interface options. Note: allowed values are `force_via_address` and `no_rport`.
TOS	This sets "type of service" field in IP packets header. Default value: 104
Greylist	Enables usage of greylist filter.

Resolver Nameservers	DNS nameservers to use while communicating through this interface. Each unique nameserver configuration has its own reverse-dns cache. If the parameters of two configurations are the same (i.e. regardless of the order, the same set of nameservers & bind-to-ip addr. flag resolves to the same physical interface), then they share a common reverse-dns cache. This rule covers the resolver configuration in the call-agents as well. Requests inbound from this interface will attempt to use the resolver configuration of this interface for DNS requests, until a call-agent is chosen. After that, if the call-agent has a resolver configuration, it will override this. When trying to find a source call-agent that is identified by DNS, a DNS reverse-cache lookup is done using the source IP. This look-up follows these steps until a match is found: 1. A reverse-cache search is done on the resolver of each call-agent that is assigned to the signaling interface that the SIP message came from. If such a call-agent does not have a nameserver configuration, then the look-up is done on the system-level resolver for that call-agent. 2. A reverse-cache search is done on the resolver of the signaling interface. If the signaling interface does not have a nameserver configuration, then the look-up is done on the system-level resolver. The same look-up logic applies to finding a destination call-agent as well.
Resolver Bind To Interface	Strictly use the underlying physical interface to send the DNS requests.

8.1.4 WebSocket signaling

The websocket application allows signaling communication over websocket interface.

If “TLS enabled” is set, a TLS profile is required.

The application is exclusive and mandatory to `WS` interface.

Parameter Name	Description
Port	Listening port of the websocket server.
TLS enabled	Enable secure communications.
Interface Options	Special interface options. Note: value must start by `wspath_`.
Greylist	Enables usage of greylist filter.
TCP keep-alive	Set TCP keep-alive value (seconds) on WS. 0 disables it. I.e. if it is set to `120`, then the SBC will try to send a TCP keep-alive after 120 seconds of inactivity and wait another 120 seconds for a response. This will happen `probes` (below) times before timing out the connection.
TCP keep-alive probes	How many times to try to send keep-alive message without getting a response.

8.1.5 HA vrrp and call state replication

Please note that the application only have effect if HA is configured and used.

The redis HA replication application uses internal redis protocol for it’s communications.

The application is exclusive and mandatory to `IMI` interface.

Parameter Name	Description
Port	Port on which call state redis will be listening. Note: value not editable (`6379`).
Enable TLS	Make use of the interface' TLS profile to authenticate and secure redis HA. Redis internal protocol is used for communications. Please note that, if used, the TLS certificate **must** either be loaded with a matching CA certificate or be registered by the node's system CA (currently latest debian:12). Note 1: disable by default. Note 2: incompatible with the "default certificate" due to the CA certificate requirement. Note 3: TLS profiles' "Verify peer certificate" option isn't taken into account.

8.1.6 Probe management

FRAFOS is preparing to launch a new product: a passive monitoring probe designed for VoIP environments. This new solution delivers VoIP monitoring capabilities comparable to those offered by the ABC SBC, generating detailed events based on observed SIP signaling and RTP traffic without actively interfering with network flows. Designed for maximum deployment flexibility, the passive monitoring probe can be operated as a lightweight container anywhere within the VoIP infrastructure. It captures live SIP messages and RTP packets, producing familiar event streams and traffic captures (including PCAP files), consistent with the output of the ABC SBC and Monitor systems.

To support the integration of probe nodes, the CCM has been extended to manage them in the same manner as existing SBC nodes. A new "Node Role" field has been introduced, allowing users to specify either "sbc" or "probe" when configuring a node. At this stage, the introduction of the "probe" node type is purely preparatory and does not alter the behavior or configuration of any existing SBC nodes.

Parameter Name	Description
Port	Port for the internal http server, provides API for stats and probe's mgmt settings.
Capturing interface	Interface to capture packets from - one specific or "any" Default value: "any"
Capturing filter	Berkley packet filter for capture Default value: "port 5060"
Enable PCAP traces	Writes individual PCAP files for each call.
Blacklisted events	List of event types that should be blacklisted: event1, event2, … e.g.: msg-probe, other-failed, other-timeout, other-ok, parse-error. Blacklisted events will not be generated.
VXLAN ports	UDP destination ports used for vxlan (tunnel endpoint). Packets arriving on these ports will be automatically decapsulated. An empty lists means disabled. The standard vxlan port is 4789.

Chapter 9

Command Line Reference

The administrative GUI is the preferred way of the ABC SBC. However there are cases like the initial configuration and/or automation when accessing the ABC SBC via Command Line is useful.

9.1 Configuration Management

CLI	Purpose	Reference
ccm-backup	backup ABC SBC configuration	ABC SBC Recovery Procedure
ccm-restore	restore a configuration backup	ABC SBC Recovery Procedure
sbc-init-config	Interactively configures the IP address or DNS name of the main configuration node from which the ABC SBC node will pull its configuration. The preferred method for new deployments is to use environment variables instead; see Configuring the SBC container.	Configuring the SBC container
ccm-publish-config	Activate the current SBC configuration and make it available for all nodes.	
cluster-config-export	Export configuration in JSON format.	
cluster-config-import	Import the configuration exported by cluster-config-export command.	
ccm-config	Manage CCM configuration options. It is equivalent to CCM->CCM config GUI screen.	
sbc-cfg-cli • fetch • convert • generate • apply • compile • defaults • template	Script which is used to generate and apply config on SBC side. It also can be used to get a config template for different services like sems or keepalived for example.	

9.2 User Management

CLI	Purpose	Reference
sbc-add-user	Add new GUI user or add a user to a group.	CLI User Management
sbc-del-user	Remove a GUI user or remove a user from a group.	CLI User Management

sbc-list-groups	Get list of existing user groups	CLI User Management
sbc-list-users	Get list of SBC users	CLI User Management
sbc-user-passwd	Change password of SBC user, unlock user locked by too many login attempts or reset two-factor authentication secret.	CLI User Management

9.3 Low-Level CLI

CLI	Purpose	Reference
sbc-loglevel action [loglevel]	Shows or sets the logging level for the ABC SBC signaling process. Action is either 'get' to retrieve current value or 'set' to set it. Loglevel takes category and level. Log files are stored in the directory /var/log/frafos	Reference of Log Level Parameters

9.4 HA CLI

In previous ABC SBC releases up to 4.1, the high availability solution used was based on Pacemaker. The ABC SBC 4.2 was a transitional release that removed the Pacemaker based HA solution, before new Keepalived based HA solution was introduced in 4.3 release.

CLI	Purpose	Reference
sbc-ha-offline	Forces the node when run to be put forcibly into HA FAULT state	
sbc-ha-online	Clears the forcibly set HA FAULT state set by sbc-ha-offline	
sbc-ha-status	Shows the node's current HA status, which can be Unknown, MASTER, BACKUP, FAULT and STOP.	High Availability statuses

Chapter 10

Reference of Used Open-Source Software

The key components of ABC SBC are built as commercial software fully owned by FRAFOS GmbH and its subsidiaries. Additionally it relies on the Linux operating systems and numerous accompanying libraries and components provided by third parties under the following license terms:

- bash , GPLv3+
- boost: Boost Software License & similar (http://www.boost.org/users/license.html)
- cronie , MIT and BSD and ISC and GPLv2+
- crontabs , Public Domain and GPLv2
- dialog , LGPLv2
- dmidecode , GPLv2+
- ethtool , GPLv2
- expat (XML parser): MIT https://sourceforge.net/p/expat/code_git/ci/master/tree/expat/COPYING
- fence-agents-all , GPLv2+ and LGPLv2+
- flite , X11-like http://www.festvox.org/flite/doc/flite_2.html
- hiredis , BSD https://github.com/redis/hiredis/blob/master/COPYING
- iLBC: BSD-like
- js , GPLv2+ or LGPLv2+ or MPLv1.1
- json-c: MIT (https://github.com/json-c/json-c/blob/master/COPYING)
- jsonxx: MIT? (https://github.com/hjiang/jsonxx/blob/master/LICENSE)
- libbcg729: GPLv3 (https://github.com/BelledonneCommunications/bcg729/blob/master/LICENSE.txt)
- libcap , LGPLv2+
- libcurl: MIT/X derivate license https://curl.haxx.se/docs/copyright.html
- libevent: BDS-like http://libevent.org/LICENSE.txt
- libisac: WebRTC license
- libopus: BSD
- libosip2 , LGPLv2+
- libpcap , BSD with advertising
- librsvg2 , LGPLv2+
- libsrtp , BSD-like https://github.com/cisco/libsrtp/blob/master/LICENSE
- libtiff , BSD-like (http://www.libtiff.org/misc.html)
- libxml2 , MIT http://www.xmlsoft.org/FAQ.html
- mailx , BSD with advertising and MPLv1.1
- mariadb-server , GPLv2 with exceptions and LGPLv2 and BSD
- monit , AGPLv3
- mysql++ , LGPLv2
- mysql-connector-c++ , GPLv2 with exceptions
- MySQL-python , GPLv2+
- nginx, BSD-like
- net-snmp , BSD http://www.net-snmp.org/about/license.html
- net-snmp-utils , BSD
- ntp , (MIT and BSD and BSD with advertising) and GPLv2
- opencore-amr: Apache V2.0

- openssh-clients , BSD
- openssl, BSD-like https://www.openssl.org/source/license.html
- opus , BSD
- pciutils , GPLv2+
- pcmisc , GPLv2+
- pcs , GPLv2
- perl-Net-SSLeay , OpenSSL
- php-cli , PHP and Zend and BSD
- php-db , PHP
- php-log , PHP
- php-mysql , PHP
- php-pear-XML-RPC , PHP
- php-pecl-runkit , PHP
- php-xmlrpc , PHP and BSD
- python , Python
- python-jinja2 , BSD
- redis , BSD
- rsync , GPLv3+
- sems-gsm , public domain
- sems-speex , modified BSD
- serweb-frmwrk , GPL
- silk: BSD-like
- spandsp (g722, DTMF): LGPL
- speex , BSD
- sqlite , Public Domain
- stunnel, GPL
- syslog-ng , GPLv2+
- sysstat , GPLv2+
- tcpdump , BSD with advertising
- vconfig , GPLv2+
- yajl (JSON): ISC license https://en.wikipedia.org/wiki/ISC_license
- wireshark , GPL+

Chapter 11

Reference Userdata Parameters for AWS Instances

The behavior of the ABC SBC can be altered by Userdata passed to it during instance launch. See the following link for more information about Userdata: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-instance-metadata.html#instancedata-add-user-data

The ability to alter the instance behavior is often useful when instances are started using a CloudFormation template. The parameters passed through Userdata must be encoded as attribute name:value pair; name and value are separated by comma and so are the pairs.

The following table shows reserved attribute names and how they are used.

Attribute Name and Value	Description
configurl <URL>	Download an ABC SBC backup configuration (only applicable when ismater:TRUE, if the instance is backup it retrieves the configuration from its master.
cwgroup <NAME>	an additional CloudWatch Dimension to which the ABC SBC sends CloudWatch metrics; this can be used to group metrics from multiple instances; note that proper CloudWatch permissions must be set
cwregion <REGION>	if CloudWatch metrics is to be gathered in a different region than instance's own, set the CloudWatch region using this parameter
ismaster TRUE	Enforce configuration master role
master <IP address>	Run this instance as configuration backup of a master identified by an IP address.
remotebootscript <URL>	URL of a bash script that will be downloaded and sourced during instance launch. The script must be finite because the boot process doesn't continue until it completes.
rtcecdns <IP address>	Address of the primary Monitor

Note that any attribute names including custom ones can be passed via Userdata. When a remotebootscript is used and started, all the attributes are passed to it as shell variables.

An example of UserData may look like this:

```
rtcecdns,172.12.1.1, configurl, https://s3-eu-west-1.amazonaws.com/frafos-abcconfig/40014-honeypot.
sql
```

Chapter 12

Reference XML-RPC functions

In the case that the ABC SBC administrator needs to configure large data sets to CCM GUI, it will be easier to provision those data automatically with a script as opposed to typing it in using the web-interface. This can be accomplished using the ABC SBC's XML-RPC data provisioning interface.

The following example shows a python code fragment for accessing the built-in XML-RPC provisioning server:

```
#!/usr/bin/python
from xmlrpc import client
server = client.Server('https://username:password@10.0.0.10:1443/rpc.php')
```

Note that for the python client, a question mark (?) in the password does not work. The user accessing XML-RPC interface has to be either member of **SBCrpc** group or member of another group having **XML-RPC** privilege.

For the XML-RPC access the IP address of the configuration master node has to be used. The XML-RPC is accessible by default on port 1443.

The XML-RPC interface is self documented via function `rpc.help()`. When the function is called without any argument it prints list of all available function. When function name is given as an argument to this function (`rpc.help(<function name>)`) it will return detailed help of the specified function.

For example try following calls in python:

```
print(server.rpc.help())
print(server.rpc.help('rpc.help'))
```

As of now functions for manipulate following entities are available:

- Provisioned Tables
- Call agents
- TLS profiles
- Nodes
- Logical interfaces
- System interfaces
- Maintenance mode

Bellow is list of all available XML RPC functions. Call `rpc.help(<function name>)` to get detailed help of specified function.

12.1 Provisioned Tables

Functions for define provisioned tables and manipulate data in them.

Function Name	Description
tables.fetch_rules($table_name, $start, $count, $key_values)	Get all rules from specified provisioned table.
tables.fetch_rule($table_name, $key_values)	Get a rule matching the key from the specified provisioned table.

tables.insert_rule($table_name, $data)	Insert rule into specified provisioned table.
tables.insert_rules($table_name, $rules)	Insert multiple rules into specified provisioned table.
tables.update_rule($table_name, $data)	Update rule of specified provisioned table.
tables.update_rules($table_name, $rules)	Update multiple rules of specified provisioned table.
tables.insert_update_rule($table_name, $data)	Try update rule of specified provisioned table. If rule with matching UUID or key columns does not exists, new rule is inserted.
tables.delete_rule($table_name, $uuids)	Delete rule(s) from specified provisioned table.
tables.delete_all_rules($table_name)	Delete all rules from specified provisioned table.
tables.commit($table_name, $msg)	Commit working version of provisioned table into use by signaling and create new working version by copying the current one.
tables.fetch()	Get all provisioned table definitions.
tables.insert($payload)	Insert provisioned table.
tables.update($payload)	Update provisioned table.
tables.delete($table_name)	Delete provisioned table.
tables.delete_room($room_name)	Delete a conference room (PIN provtable type).

For example, to introduce a new entry to the blacklist and check the outcome, the following three RPC commands must be called: *insert_rule*, *commit* and *fetch_rules*:

```
data = {"key_value":"sip:restricted@abc.com"}
print(server.tables.insert_rule('test_uri_bl',data))
print(server.tables.commit('test_uri_bl', 'new restricted used introduced'))
print(server.tables.fetch_rules('test_uri_bl'))
```

This script will result in the following list of URIs shown on the command-line output:

```
[{'key_value': 'sip:banned@abcsbc.com', 'uuid': '6c01a834-9d32-df09-0217-000000f074ee'},
{'key_value': 'sip:forbidden@abcsbc.com', 'uuid': '54d15a12-62bc-73c9-8313-000012f8ae1b'},
{'key_value': 'sip:restricted@abc.com', 'uuid': '6d831a12-88bc-7fa9-7483-000083ff992a'}]
```

Note that the routing tables have several predefined mandatory elements that must use the following conventions:

- *cagent* takes name or UUID of a call-agent
- *outbound_proxy* and *next_hop* is passed as string
- boolean parameters *next_hop_1st_rq*, *upd_ruri_host*, and *upd_ruri_dns_ip* take either 0 or 1 as value
- the enumerative parameters *route_via* takes one of the following values: *outbound_proxy*, *next_hop* or *ruri*

12.2 Call agents

Function Name	Description
cagents.fetch($filter)	Get call agents
cagents.insert($payload)	Insert call agent
cagents.update($payload)	Update call agent
cagents.delete($realm_name, $cagent_name)	Delete call agent

cagents.add_target($realm_name, $cagent_name, $payload)	Add target destination to call agent
cagents.del_target($realm_name, $cagent_name, $payload)	Remove target destination from call agent

12.3 TLS profiles

Function Name	Description
tls_profile.fetch($filter)	Get TLS profiles
tls_profile.insert($payload)	Insert TLS profile
tls_profile.update($payload)	Update TLS profile
tls_profile.delete($name)	Delete TLS profile

12.4 Nodes

Function Name	Description
node.fetch($filter)	Get SBC nodes
node.insert($payload)	Insert SBC node
node.update($payload)	Update SBC node
node.delete($name)	Delete SBC node

12.5 Logical interfaces

Function Name	Description
log_interface.fetch($filter)	Get logical interfaces
log_interface.insert($payload)	Insert logical interface
log_interface.update($payload)	Update logical interface
log_interface.delete($name)	Delete logical interface
log_interface.help_app_list()	Return list of available applications
log_interface.help_app($application)	Return detailed info about an application

12.6 System interfaces

Function Name	Description
sys_interface.fetch($filter)	Get system interfaces
sys_interface.insert($payload)	Insert system interface
sys_interface.update($payload)	Update system interface
sys_interface.delete($log_if_name, $owner_type, $owner_name)	Delete system interface

12.7 Maintenance mode

If the “maintenance mode” is activated, the SBC answers 503 to any request.

The XMLRPC interface allows to toggle a "maintenance mode" for a given node. Please use `sems-stats -c "set_shutdown 1"` to trigger the maintenance mode, or `sems-stats -c "set_shutdownmode 0"` to disable it. At any time, one may use `sems-stats -c "get_shutdownmode"` to fetch the current node status.

One may also trigger the maintenance mode via the `gopi` API (:4224), using either the `/api/v1/enable/shutdownmode` or the `/api/v1/disable/shutdownmode` endpoints.

Finally, a helper script `sbc-shutdownmode` exists. Please refer to `sbc-shutdownmode -h` for more information about it.

Chapter 13

Reference of CCM Configuration Parameters

This reference lists all CCM configuration parameters. The configuration parameters are grouped as follows:

- Login
- LDAP Parameters
- Backup Parameters
- Management access Parameters
- SBC security Parameters
- Email Parameters
- Certbot Parameters
- Miscellaneous Parameters

Each section specifies an API category and each attribute has the *API* name listed in its description. Use these two names together to set or retrieve a value through the CCM REST API.

13.1 Login

Parameters related to login/logout.

API category: login

Parameter Name	Description
GUI auto-logout time	Timeout (in minutes) of inactivity after which the GUI user is automatically logged out. If changed, relogin is required. Use '0' to disable auto-logout. Default value: 15 API: gui_logouttime
Max failed login	Maximum number of failed login attempts before the user account is blocked. This protects against brute-force attacks. Use '0' to disable account blocking. Default value: 3 API: login_max_failed
Blocking period	Duration (in seconds) that the user account remains blocked after reaching the "Max failed login" limit. Default value: 300 API: login_blocking_period
Allow concurrent login	Allows a single GUI user to log in from multiple devices simultaneously. Disabled by default. Default value: enabled API: login_allow_concurent

Parameter Name	Description
Garbage collect timeout	Timeout (in days) after which brute-force protection data is removed from the database. Default value: 180 API: login_gc
Do not allow re-use passwords - history length	Prevents users from setting a password identical to any of the last N passwords used. This field specifies N. Use '0' to disable history length. Default value: disabled API: password_history
Password expiration	Number of days before the user password expires and must be changed. Use '0' to disable expiration. Default value: disabled API: password_expire
Minimum password length	Minimum required length of a user password. Default value: 8 API: password_min_length
Password strength policy	Defines the set of character types that must be included in a user password. Default value: At least one uppercase letter, one lowercase letter, and one number. API: password_strength Possible values over API: • "": No policy • "aA": At least one uppercase and one lowercase letter • "aA0": At least one uppercase, one lowercase letter and one number • "aA0.": At least one uppercase, one lowercase letter, one number and one special character
Enable two-factor authentication for LDAP users (initial value)	Enables two-factor authentication for LDAP users upon their first connection to the CCM. Default value: disabled API: login_ldap_totp_enabled
Relying Party ID for passkeys	Identifier of the WebAuthn Relying Party on whose behalf registration or authentication is performed. A credential can only be used with the same RP ID it was registered with. By default, the RP ID is set to the caller's origin's effective domain. For more information see w3.org. Default value: empty API: login_webauth_rp_id

Table 25: Login Parameters

13.2 LDAP Parameters

The Cluster Config Manager GUI supports two-step authentication against an LDAP server.

The first authentication step verifies a user against the LDAP server. In this step, the user DN (e.g., `uid=john,ou=People,dc=example,dc=org`) and password (e.g., `johnldap`) are used.

The second step ensures that at least one of the LDAP user's groups matches one of the existing GUI groups.

Once configured, users can log in to the Cluster Config Manager using their LDAP UID and password on the login page.

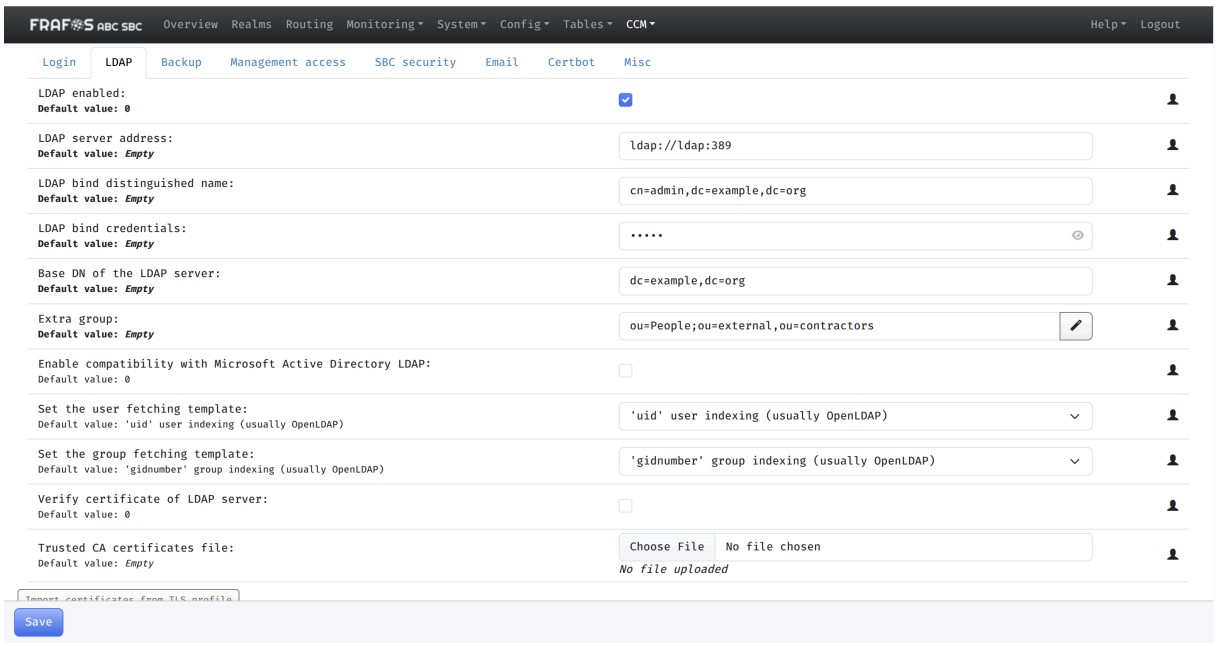
API category: ldap

Parameter Name	Description
LDAP enabled	Enables or disables LDAP authentication. Default value: disabled API: ldap_enabled
LDAP server address	Address of the LDAP server in the format `ldap://IP:PORT`, `ldap://IP`, or `ldap://my.domain`. Default value: empty API: ldap_host
LDAP bind distinguished name	Distinguished Name (DN) used to bind to the LDAP server for lookups. Default value: empty API: ldap_user
LDAP bind credentials	Credentials (password) for the LDAP bind user. Default value: empty API: ldap_password
Base DN of the LDAP server	Default search DN for the LDAP directory. Example: for `cn=admin,dc=example,dc=org`, the base DN is `dc=example,dc=org`. Default value: empty API: ldap_base_dn
Extra group	Additional bind DN part for user entries. For example, if a user `john` exists as `uid=john,ou=People,dc=example,dc=org`, set this parameter to `ou=People`. The system will build the full DN as `uid=[username],[extra_group],[base_dn]`. You can enter multiple groups by separating them with a semicolon. Default value: empty API: ldap_extra_group
Enable compatibility with Microsoft Active Directory LDAP	Enables compatibility mode for Microsoft Active Directory LDAP servers. Default value: disabled API: ldap_win

Parameter Name	Description
Set the user fetching template	Select the appropriate LDAP attribute for usernames. For Microsoft Active Directory, use `sAMAccountName`. For typical OpenLDAP setups, use `uid`. Some setups may require `cn`. Default value: 'uid' user indexing (usually OpenLDAP) API: ldap_user_template Possible values over API: • "(cn=%(username)s)" • "(uid=%(username)s)" • "(sAMAccountName=%(username)s)": Microsoft Active Directory
Set the group fetching template	Select the appropriate LDAP attribute for groups. For Microsoft Active Directory, use `memberOf`. For typical OpenLDAP setups, use `gidNumber`. Some may require `memberUid`. Default value: 'gidnumber' group indexing (usually OpenLDAP) API: ldap_group_template Possible values over API: • "gidNumber" • "memberUid" • "memberOf": usually for Microsoft Active Directory
Verify certificate of LDAP server	If enabled, the LDAP server's TLS certificate is verified. Default value: disabled API: ldap_reqcert
Trusted CA certificates file	Path to a PEM-formatted file containing trusted CA certificates used to verify the LDAP server's certificate. Default value: empty API: ldap_tlscacert

Table 26: LDAP Parameters

13.2.1 OpenLDAP configuration example



There is a docker container available on github that match the screenshot configuration : https://github.com/frafos/docker-ldap.

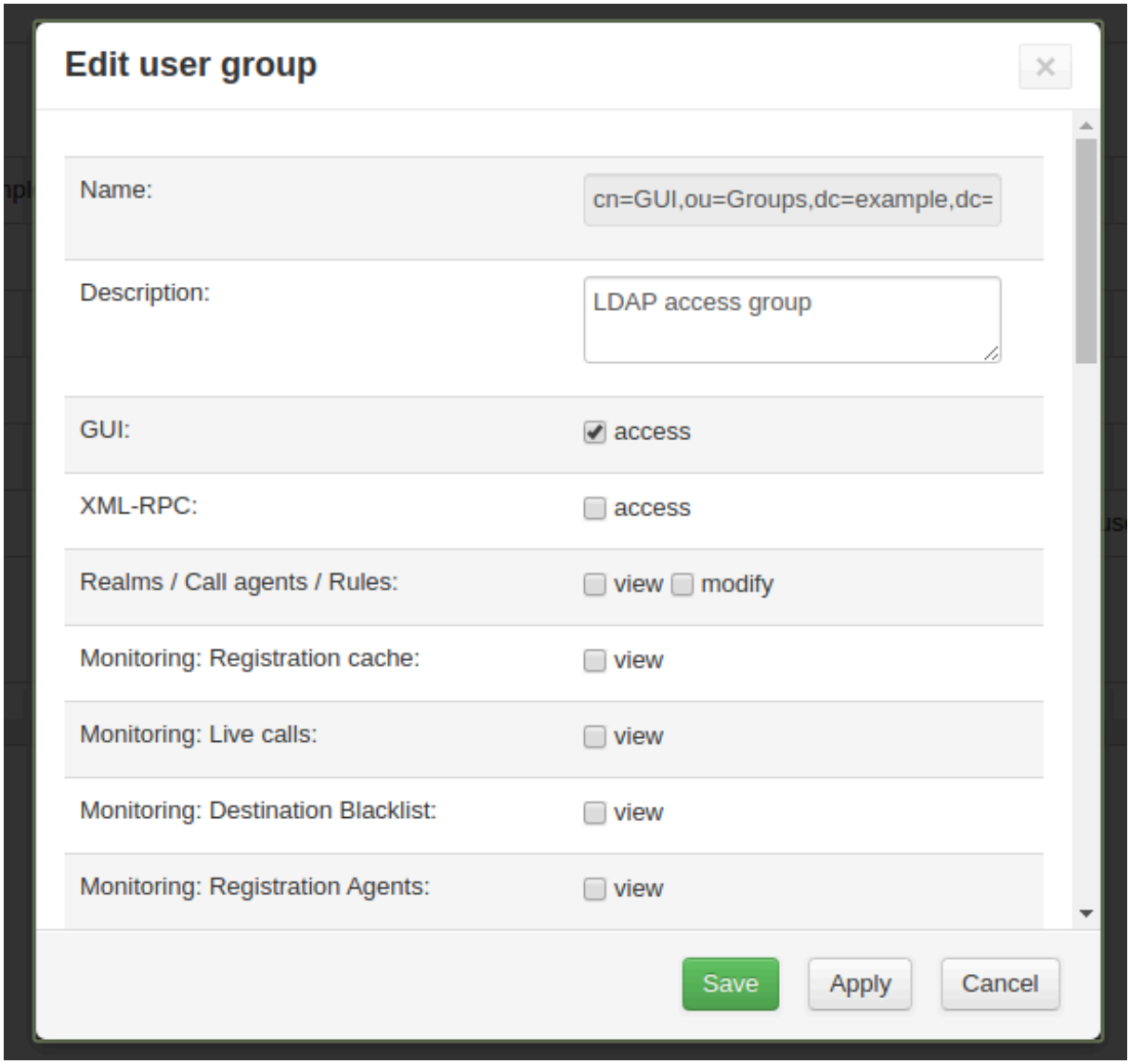
The image come in with 3 users (+ admin) :

User	dn	pwd	note
john	`uid=john,ou=People, dc=example,dc=org`	`johnldap`	The following example works for this user.
jane	`uid=jane,ou=People, dc=example,dc=org`	`janeldap`	The following example **doesn't** work for this user. John and Jane belong to different groups.
june	`uid=june,ou=external, ou=contractors, dc=example,dc=org`	`juneldap`	The following example works for this user.

Table 27: OpenLDAP users

In that following ldap, user `john` can be authenticated against the ldap via `uid=john,ou=People,dc=example,dc=org`. To allow an ldap user to access the ABC SBC GUI, a **GUI group name** with access to the GUI **must** match one of the primary group of the ldap user.

So we create GUI group named after the full dn of one `john` LDAP group (`cn=GUI,ou=Groups,dc=example,dc=org`) :



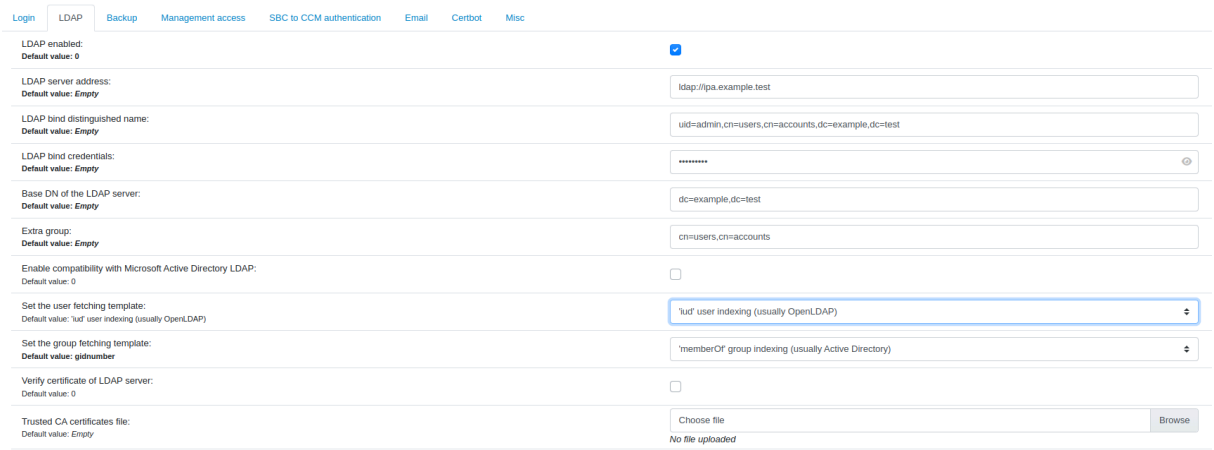
You can then login with the credential `john` and the password `johnldap`.

Note that the following credentials are also valid:

- `june/juneldap`

Note: If we want Jane to be able to access the GUI, we'll need to define another ABC SBC GUI groups, matching one of Jane ldap groups name (`cn=Mistyc,ou=Groups,dc=example,dc=org` in this case).

13.2.2 FreeIPA LDAP configuration example



We'll skip the server configuration part for sanity reasons. We recommend to have a look at https://www.freeipa.org/page/Docker for easy setups.

In the following case, the Free IPA server was configured with default values, generating the following configuration:

```
The IPA Master Server will be configured with:
Hostname:       ipa.example.test
IP address(es): 172.42.0.142
Domain name:    example.test
Realm name:     EXAMPLE.TEST

The CA will be configured with:
Subject DN:   CN=Certificate Authority,O=EXAMPLE.TEST
Subject base: O=EXAMPLE.TEST
Chaining:     self-signed

Client hostname: ipa.example.test
Realm: EXAMPLE.TEST
DNS Domain: example.test
IPA Server: ipa.example.test
BaseDN: dc=example,dc=test
```

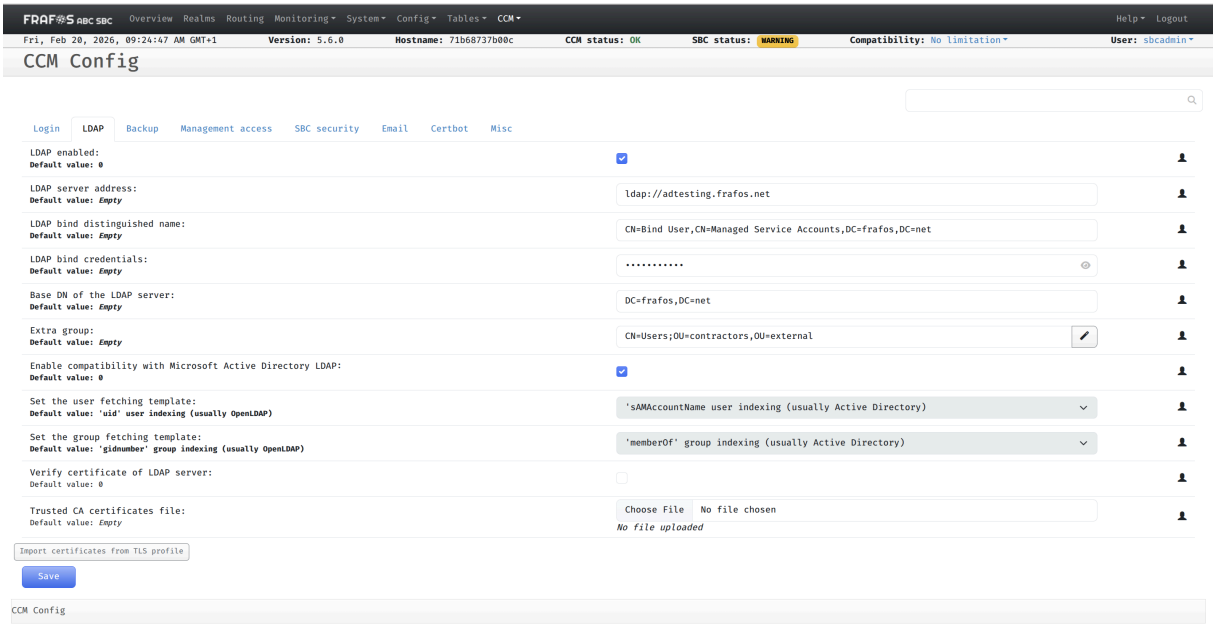
On the FreeIPA side, we've created an `sbcgui` group and a `john` user belonging to that group. We can query them over the ldap with the following:

```
$ ldapsearch \
  -D "uid=admin,cn=users,cn=accounts,dc=example,dc=test" \
  -w [admin password] \
  -H ldap://ipa.example.test \
  -b dc=example,dc=test 'uid=john'
(...)

# john, users, accounts, example.test
dn: uid=john,cn=users,cn=accounts,dc=example,dc=test
givenName: John
sn: Doe
uid: john
cn: John Doe
displayName: John Doe
initials: JD
gecos: John Doe
krbPrincipalName: john@EXAMPLE.TEST
gidNumber: 681800003
objectClass: top
objectClass: person
objectClass: organizationalperson
objectClass: inetorgperson
objectClass: inetuser
objectClass: posixaccount
objectClass: krbprincipalaux
objectClass: krbticketpolicyaux
objectClass: ipaobject
objectClass: ipasshuser
objectClass: ipaSshGroupOfPubKeys
objectClass: mepOriginEntry
objectClass: ipantuserattrs
loginShell: /bin/sh
homeDirectory: /home/john
mail: john@example.test
krbCanonicalName: john@EXAMPLE.TEST
ipaUniqueID: c81b0b0e-950a-11ee-8471-0242ac2a008e
uidNumber: 681800009
krbPasswordExpiration: 20231207141322Z
krbLastPwdChange: 20231207141322Z
krbExtraData:: AAIC03Flcm9vdC9hZG1pbkBFWEFNUExFLlRFU1QA
mepManagedEntry: cn=john,cn=groups,cn=accounts,dc=example,dc=test
ipaNTSecurityIdentifier: S-1-5-21-1615603866-3760360139-3083941652-1009
memberOf: cn=ipausers,cn=groups,cn=accounts,dc=example,dc=test
memberOf: cn=sbcgui,cn=groups,cn=accounts,dc=example,dc=test
```

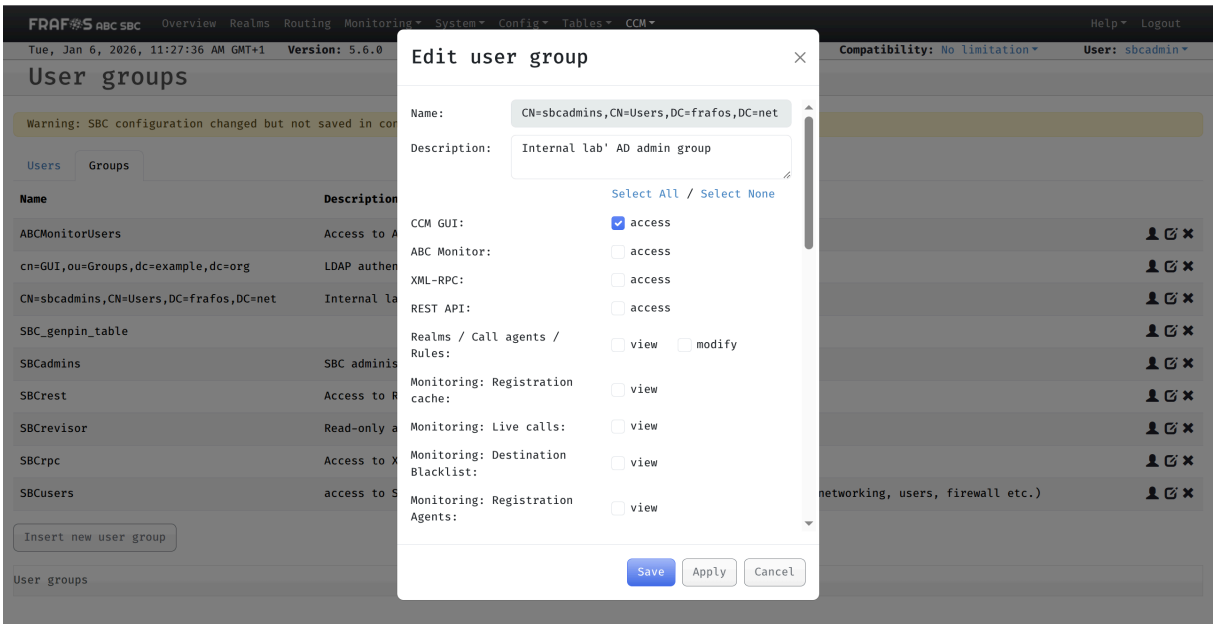
On the CCM side, we've created a new `cn=sbcgui,cn=groups,cn=accounts,dc=example,dc=test` group with GUI permissions. Please note that the CCM group matches the LDAP full CN.

13.2.3 Microsoft Active Directory configuration example



Active Directory configuration slightly differs from the OpenLDAP one, please note that all value are **case-sensitive**!

On the CCM side, we've created a new `CN=sbcadmins,CN=Users,DC=frafos,DC=net` group (this is the full name of the group) with appropriate GUI permissions. Please note that the CCM group matches the LDAP full CN.



13.3 Backup Parameters

These parameters configure ABC SBC daily backups. For more information, see Backup and Restore Operations.

API category: backup

Parameter Name	Description
Create daily SBC configuration backups	If enabled, a daily snapshot of the SBC configuration is created as a gzipped tarball file. Automatic backups are created when a new container starts and before database upgrade is performed. This backup file allows a rollback to a previous container if required. Default value: disabled API: backup_enable
Include provisioned tables in daily or automatic backups	If enabled, daily or automatic backups also include the full content of provisioned tables. Default value: enabled API: backup_prov
Number of days to keep backups	Defines the retention period (in days) for backup files. All files matching `sbc-backup-*` in the backup directory that are older than the specified number of days are deleted during the daily backup process. Use `0` to disable automatic deletion. Default value: 7 API: backup_keep
Destination directory for backups	Specifies the target directory for daily backup files. Default value: `/data/backups` API: backup_dir

Table 28: Backup Parameters

13.4 Management access Parameters

API category: xmi

Parameter Name	Description
SSL certificate file for GUI, REST API and XML-RPC interface	A file containing SSL certificate in PEM format, used for the GUI, REST API, and XML-RPC interfaces. Default value: empty API: tlscert
SSL private key file for GUI, REST API and XML-RPC interface	A file containing a private key in PEM format, corresponding to the SSL certificate, used for the GUI, REST API, and XML-RPC interfaces. Default value: empty API: tlskey
Minimal supported TLS version for GUI, REST API and XML-RPC interface	Specifies the minimum TLS version to be supported on the GUI, REST API, and XML-RPC interfaces. Default value: tls1.2 API: tls_min_version

Table 29: Management access Parameters

13.5 SBC security Parameters

These parameters are used for securing the connection between SBCs and the CCM for services running on the IMI interface.

API category: pullconf

Parameter Name	Description
HTTP Basic Authentication username for configuration pull or status push	Username that SBC nodes use to pull the configuration from the CCM or to push status. The same value must be provided to the SBC container via the `CONFIG_USER` environment variable (see Configuring the SBC container). API: user
HTTP Basic Authentication password for configuration pull or status push	Password that SBC nodes use to pull the configuration from the CCM or to push status. The same value must be provided to the SBC container via the `CONFIG_PASS` environment variable (see Configuring the SBC container). API: pass
SSL certificate file for pullconf	A file containing the SSL certificate in PEM format. Default value: empty API: tlscert
SSL private key file for pullconf	A file containing the private key for the SSL certificate in PEM format, unencrypted (no passphrase protection). Default value: empty API: tlskey
Trusted CA certificates file	A file containing the list of trusted CA certificates against which clients' TLS certificates are verified. If intermediate CAs are used, include the full chain. Certificates must be in PEM format. Default value: empty API: tlscacert
Enable mTLS	If enabled, the CCM verifies the peer's TLS certificate against the trusted CA certificates. Default value: disabled API: mandate_client_cert
SBC API authentication token	Bearer token used by the Cluster Config Manager to authenticate against the ABC SBC node API. This token is auto-generated by the Cluster Config Manager and distributed to all ABC SBC nodes as part of their configuration. Use the **Renew** button to invalidate and regenerate the token. After confirmation, the Cluster Config Manager automatically pushes the new configuration to nodes configured in push mode. Nodes in pull mode will pick up the new token on their next pull. The previous token is kept as a fallback for nodes that have not yet received the update. Auto-generated if not set. Not accessible via API.

Table 30: SBC security Parameters

13.6 Email Parameters

These parameters are used to configure sending emails from the Cluster Config Manager.

API category: email

Parameter Name	Description
Email address for sending certificate and other alerts	Email address to which important alerts, such as certificate renewal failure or acquisition success, are sent. This field must be set if any Let's Encrypt certificate is expected to be used. Default value: empty API: alert_email
From email address for sending alerts	Email address used as the "From" address in alert emails. If empty, the system default is used. Default value: empty API: alert_email_from
SMTP email server address for sending alerts	SMTP server address used by CCM to send alert emails. Default value: empty API: alert_mail_server
SMTP mail server port	SMTP mail server port. Default value: 25 API: alert_mail_server_port
Use secure connection to SMTP mail server	Controls whether the SMTP connection should be encrypted using TLS or STARTTLS. Default value: no API: alert_mail_server_tls
SMTP mail server authentication	Set to `off` to disable authentication, or `on` to enable it and select the auth type automatically. Default value: off API: alert_mail_server_auth
Username for SMTP authentication	Username for SMTP authentication, if authentication is enabled. Default value: empty API: alert_mail_server_user
Password for SMTP authentication	Password for SMTP authentication, if authentication is enabled. Default value: empty API: alert_mail_server_pass

Table 31: CCM Email Parameters

13.7 Certbot Parameters

Cluster Config Manager certbot acts like the standard Let's Encrypt certbot. For more information, see the TLS chapter: Let's encrypt gocertbot.

API category: certbot

Parameter Name	Description
Query Let's Encrypt staging environment	If testing, query the Let's Encrypt staging environment to avoid hitting the Let's Encrypt 168-hour rate limit. Note: Staging certificates are not suitable for production use. Default value: disabled API: certbot_query_stagging
Attempt renewal X days before certificate expiration	Number of days before certificate expiration when certbot attempts renewal. This setting does not affect automatic email notifications about certificate expiration sent by Let's Encrypt. Default value: 15 API: certbot_days_renew
CRON job interval	CRON schedule (in CRON format) defining how often certbot runs to attempt renewal of certificates near expiration. Default value: 0 1 * * * API: certbot_cron_interval (the valid value is for example: `"0 1 * * *"`)
Preferred CA chain	If the CA provides multiple certificate chains, prefer the one whose issuer matches the specified Subject Common Name. If no match is found, the default offered chain will be used. Default value: ISRG Root X1 API: certbot_preferred_chain

Table 32: Certbot Parameters

The certbot is invoked automatically under the following conditions:
- By a CRON job, every night at 01:00.
- When a node successfully pulls a new configuration.
- When a configuration is successfully pushed to a node.

You can also invoke certbot manually from within a Cluster Config Manager shell by running:

```
% sbc-gocertbot -d
```

Note: For testing purposes, to avoid reaching the Let's Encrypt 168-hour rate limit, enable the *Query Let's Encrypt staging environment* option in the Cluster Config Manager configuration.

13.8 Miscellaneous Parameters

API category: misc

Parameter Name	Description
Automatically add new nodes	If enabled, records for new nodes that pull configuration from the configuration master will be automatically added. If disabled, the CCM will provide configuration only to nodes that are already registered in the CCM. Not yet registered nodes will be refused. Default value: enabled API: auto_add_nodes

Parameter Name	Description
Compatibility mode	If using CCM with older SBCs, the SBC version can be selected here. CCM will hide settings (e.g., rule conditions and actions, interface applications, global config values, or entire screens) that are not available in the selected SBC version. Selecting the “No limitation” option displays all settings for all supported SBC versions. Default value: <the latest SBC version> API: compatibility_mode (use “” for “No limitation”, or SBC version in the format “X.Y” - for example “5.6”)
Compatibility mode with secunet SBC	If enabled, the firewall control and HA configuration screens will be hidden. Default value: disabled API: secunet_compat_mode
Allow overlap of Call Agent IP ranges	If enabled, the GUI will not check whether IP address ranges of call agents overlap. Default value: disabled API: allow_ca_overlap
Address of ABC Monitor GUI	If set, a link to the ABC Monitor GUI will be added to the top bar menu under the Monitoring tab. If not set, the value of the ABC Monitor address from Event Parameters is used. Default value: empty API: monitor1_gui_url
Address of secondary ABC Monitor GUI	If set, a link to the secondary ABC Monitor GUI will be added to the top bar menu under the Monitoring tab. If not set, the value of the Secondary ABC Monitor address from Event Parameters is used. Default value: empty API: monitor2_gui_url
Name of the CCM instance	Name of this CCM instance that will be displayed in the browser tab. If not set, “CCM” will be used. Default value: empty API: ccm_name

Table 33: Miscellaneous Parameters

Chapter 14

Reference of Token Capabilities

CCM API tokens and SBC API tokens use separate sets of capabilities. This reference lists the capabilities available for each token type and describes how SBC token capabilities map to Cluster Config Manager user permissions.

14.1 CCM API token capabilities

CCM API tokens authenticate against the Cluster Config Manager REST API. Their capabilities mirror the Cluster Config Manager user permission model. When creating or editing an API token, only capabilities and actions that the current user holds are available for selection.

The `restapi:access` capability is always added to API tokens automatically.

For the full list of Cluster Config Manager user capabilities, see the user group configuration in *CCM –> CCM config –> Groups*.

14.2 SBC API token capabilities

SBC API tokens authenticate against the ABC SBC node API. They use a dedicated set of capabilities that control access to SBC node operations.

Capability	Action	Description
sbc-config	read	Read SBC configuration
sbc-config	push	Push configuration changes to the SBC node
sbc-provtables	read	Read provisioning tables
sbc-provtables	push	Push provisioning table changes to the SBC node
sbc-status	read	Read SBC status and monitoring data
sbc-system	exec	Execute system administration commands

Table 34: SBC token capabilities

14.2.1 Permission mapping

When creating or editing an SBC token, the selectable capabilities are limited by the Cluster Config Manager user's own permissions. The following table shows which Cluster Config Manager user capability and action is required to assign each SBC token capability.

SBC Capability	SBC Action	Required CCM User Capability	Required CCM Action
sbc-config	read	sbc-config (Activate SBC configuration)	activate
sbc-config	push	sbc-config (Activate SBC configuration)	activate
sbc-provtables	read	prov-tables-tbl (Tables: values)	view
sbc-provtables	push	prov-tables-tbl (Tables: values)	activate
sbc-status	read	mon-node-status (Monitoring: System status)	view
sbc-system	exec	sys-restart (System: Administration)	access

Table 35: SBC token to CCM user permission mapping

The required permission for each ABC SBC API endpoint is documented in the ABC SBC OpenAPI specification as the `x-required-permission` extension on each operation.

Chapter 15

CCM configuration API

Starting 5.5, the CCM ships a RESTful API allowing SBC nodes' JSON configuration interactions.

The API can be accessed at *http://localhost:1444*. All requests to the CCM's *:444* port, with an URI prefixed by */exportconf/*, are forward to the API.

A non exhaustive list of the available endpoints actions are:

- fetch an SBC JSON configuration for a given set of parameters (node's release/uuid, config group etc …)
- list nodes using a specific TLS profile
- list nodes using any Let's Encrypt TLS profile

The API does **not** offer any configuration to the user.

Chapter 16

Reference of Supported Codecs

This reference lists all supported codecs by ABC SBC.

- PCMU/8000
- G721/8000
- GSM/8000
- PCMA/8000
- g722/8000
- L16/32000
- L16/16000
- L16/8000
- G726-32/8000
- G726-24/8000
- G726-40/8000
- G726-16/8000
- G729/8000
- opus/48000
- isac/16000
- iLBC/8000
- speex/32000
- speex/16000
- speex/8000
- AMR/8000
- AMR-WB/16000

Chapter 17

SIP Timers

17.1 Timer Definitions

The SBC SIP stack uses timers to control the transaction state machine and parts of the dialog state machine.

In general, timers A to J are used as defined in RFC3261.

17.1.1 Timer L

Timer L is used to determine for how long a UAC transaction will stay in memory after a 2xx reply has been received. It corresponds roughly to the transaction state machine fixes described in RFC6026 as timer M.

17.1.2 Timer M

Timer M is a reduced transaction timer used to implement tranport layer failover. It defaults to a quarter (1/4) of the client transaction timer (timer B), such that up to 4 destinations can be probed before the SIP client transaction will be considered expired.

Please note that timer M is only used when there are indeed multiple destinations to be probed.

17.1.3 Timer BL

Timer BL is used to prolong the transaction timer for the purpose of properly handling destination blacklisting.

Once a transaction has expired and no reply has been received, timer BL is started. If a reply is received before it expires, the timer is cleared and the destination will not be placed into the destination blacklist.

If however no reply is received and the timer expires, the destination will be placed into the destination blacklist.

Please note that timer BL is only used if destination blacklisting is enabled for that particular transaction / destination.

17.2 Customizing Timer Values

It is to be noted that the SBC uses by default the timer values as advised by the respective RFCs. However it does not enforce that the values are set relatively to each other when setting custom values.

This means that great care needs to be taken to respect correct proportions when using values that differ from the defaults. In particular, these timers are not set dependent to T1, which is not used when configuring custom timer values.

17.3 SIP Timer Precision

The SBC uses so called hierarchical timers to implement SIP timers effeciently. This together with the inherent imprecision of timers and latency of non-real-time operating systems introduces a few limitations on the precision achievable by the timer settings.

SIP timers are triggered in slots of 20ms each, meaning that a timer expiry routine might be executed anywhere inside that 20ms slot, depending on many factors. It is thus common that very short timers suffer from this imprecision and might display a huge variance.

For example, settings any re-transmission timer to a value of 40ms may lead to the first re-transmission happening anywhere between 20 and 60 milliseconds, depending on the exact slot the timer has been placed in, the time at which it has been started, or even the overall number of timers hitting around that time.

17.4 Destination Monitoring

The destination monitoring feature uses by default a different set of SIP timers to allow for higher reactivity.

The timers used are as follows:

- Re-transmission timer (E): 2 second
- Transaction timer (F): 5 second

The values are chosen to avoid interfering with the normal traffic, however still allowing for a shorter transaction timeout to react faster to unavailability.

Chapter 18

Legacy application

18.1 SSH

> **Info**
>
> Support dropped starting 5.5.

The ssh application allows a shell access via the associated interface on the configured port options. The application may be enabled on all interface types.

Parameter Name	Description
Port	Port allowing ssh access.

18.2 SNMP

> **Info**
>
> Support dropped starting 5.5. Please use Prometheus via Unified SBC management service instead.

The snmp application enables SNMP daemon listening. Note that this application only has effect on SBC node.

It does not require TLS profile, as TLS is not used.

The application may be enabled on `CX` interface.

Parameter Name	Description
Port	Port on which the SNMP server listens.

18.3 TURN server for websocket

> **Info**
>
> Application available since SBC release 4.5.

> **Info**
>
> Support dropped starting 5.2.

It enables the TURN server on given node. It is possible to configure one TURN server per node but it can be configured for more than one node.

It does not require a TLS profile.

The application may be enabled on `CX` interface.

Note well: using the TURN server application might expose the SBC to certain security risks. Indeed, the TURN server application makes use of static credentials for compatibility purposes, such that these well known credentials might be misused. It is therefor important to limit the use of the TURN server application to the use case where it is absolutely required (support TCP media transport). Enabling this application is absolutely not necessary to supporting WebRTC in general.

Parameter Name	Description
Listening port	Listening port of the TURN server.
Aux server	Auxiliary server address in the format `IP:port`.
Relay IP	Note: mandatory.
External IP	TURN Server public/private address mapping, if the server is behind NAT. In that situation, the External IP will be reported as relay IP address of all allocations. This scenario works only in a simple case when one single relay address is be used, and no RFC5780 functionality is required. That single relay address must be mapped by NAT to the 'external' IP. The External IP value, if not empty, is returned in XOR-RELAYED-ADDRESS field. For that 'external' IP, NAT must forward ports directly (relayed port 12345 must be always mapped to the same 'external' port 12345).
UDP port range min port	Sets the UDP range that is used for relaying media start port. Note: mandatory.
UDP port range max port	Sets the UDP range that is used for relaying media end port. Note: mandatory.
Auth user	Sets the username used for TURN server authentication. Note: mandatory.
Auth password	Sets the password used for TURN server authentication. Note: mandatory.
Realm for users	Realm passed, which is usually domain name.
Media IP to allow UDP on firewall	Sets the IP address that will be allowed on SBC firewall to talk to the TURN.
UDP port range min port for media IP	Sets the UDP range that is used for media IP, start port.
UDP port range max port for media IP	Sets the UDP range that is used for media IP, end port.

18.4 Local monitoring query service

> **Info**
>
> Application available since SBC release 4.2.

> **Info**
>
> Replaced by Unified SBC management service starting 5.5.

The `sbc-xmloredis` API serves some metrics issued from different sources. The API will by default listen on the `localhost` interface, reachable via `http`. For every other interface application enabled, the API will listen exclusively via `https`, serving the configured TLS profile, which is required.

A non expose list of the available endpoints information and actions are:

- read and delete various data about registration cache
- read and delete various data about live calls
- fetch various data about monitored destination
- fetch various data about process's statistics

- support for Let's Encrypt HTTP01 challenge
- read and write various data about blacklists' call agent
- read and write various data about blacklists' destinations
- fetch various data about registration agent

The application only exists on SBC node. It is also exclusive and mandatory to `IMI` interface.

Parameter Name	Description
Port	Port on which the API server listens. Note: value not editable (`4242`).

18.5 PCAP query service

Info

Application available since SBC release 4.5.

Info

Replaced by Unified SBC management service starting 5.5.

The `sbc-pkapman` API generates and serves pcap files based on an aggregation of the pcap files available on the file system. The API will by default listen on the `localhost` interface, reachable via `http`. For every other interface application enabled, the API will listen exclusively via `https`, serving the configured TLS profile, which is required.

Requirements: SEMS's global option "Dump TLS session keys to file" Signaling SSL must be enabled if one wishes to download both pcap files and session TLS keys into a zip'ed bundle. Otherwise, the bundle may only contain pcap files.

Limitations: WebRTC interface don't support dump of the TLS keys.

A non expose list of the available endpoints information and actions are:
- fetch SBC node' file system PCAP files' timestamps
- merge SBC node' file system PCAP files as one
- merge SBC node' file system PCAP files as a ZIP with TLS keys

The application only exists on SBC node and it is mandatory and exclusive to `IMI` interface.

Parameter Name	Description
Port	Port on which the API server listens. Note: value not editable (`4243`).

18.6 Local webconf API

Info

Application available since SBC release 4.6.

Info

Replaced by Unified SBC management service starting 5.5.

The `sbc-webconf` API expose information and actions related to SEMS's web-conferencing features.

A non expose list of the available endpoints information and actions are:
- create and read rooms
- kick / mute / unmute room's active participants
- create room's dialouts
- create and edit room's pin
- read server information

The API will by default listen on the `localhost` interface, reachable via `http`. For every other interface application enabled, the API will listen exclusively via `https`, serving the configured TLS profile, which is required.

The application is exclusive and mandatory to `IMI` interface.

Parameter Name	Description
Port	Port on which the API server listens. Note: value not editable (`4244`).

18.7 Management for host

Info

Application available since SBC release 4.5.

Info

Replaced by Unified SBC management service starting 5.5.

The `sbc-goministrator` API run various administrator tasks from RESTful endpoints.

A non exhaustive list of the available endpoints actions are:
- toggle the node' HA mode
- toggle the node' maintenance node
- restart/halt/shutdown the node

The API will by default listen on the `localhost` interface, reachable via `http`. For every other interface application enabled, the API will listen exclusively via `https`, serving the configured TLS profile, which is required.

The application may be enabled on `IMI` interface.

Parameter Name	Description
Port	Port on which the API server listens. Note: value not editable (`4249`).

18.8 Log files provider

Info

Application available since SBC release 5.1.

Info

Replaced by Unified SBC management service starting 5.5.

The `sbc-goplog` API allow HTTP interactions with the SBC node' file system log files.

A non exhaustive list of the available endpoints actions are:
- list the SBC node' file system log files
- read the SBC node' file system log files by streaming the lnav application trough websockets

The API will by default listen on the `localhost` interface, reachable via `http`. For every other interface application enabled, the API will listen exclusively via `https`, serving the configured TLS profile, which is required.

The application is exclusive to `IMI` interface.

Parameter Name	Description
Port	Port on which the API server listening. Note: value not editable (`4250`).

Application is available since ABC SBC' release 5.1.

18.9 Local packet classifier

Info

Application available since SBC release 5.4.

Info

Replaced by Unified SBC management service starting 5.5.

The `sbc-gopacla` API allow to partially interact with the ABC SBC firewall. Currently, the API allow to list nftable sets entries, add entry to nftable set or prune the entry / sets.

A non exhaustive list of the available endpoints actions are:
- feed and prune the node firewall' sets and their entries
- test if entries exist in the node firewall' sets

The API will by default listen on the `localhost` interface, reachable via `http`. For every other interface application enabled, the API will listen exclusively via `https`, serving the configured TLS profile, which is required.

The application is exclusive to `IMI` interface.

Parameter Name	Description

Port	Port on which the API server listens. Note: value not editable (`4252`).

18.10 HTTP proxy

> **Info**
>
> Application available since SBC release 4.6.

> **Info**
>
> Support dropped starting 5.5.

Setup an HTTP proxy, based on nginx reverse proxy: ProxyDoc.

The application adds the `X-Real-IP`, `Upgrade` and `Connection` headers. The template (`/etc/frafos/templates/nginx/proxy.tmpl`) may be overloaded, as described in Command Line Reference.

If "TLS enable" is set, a TLS profile is required.

The application may be enabled on `CX` interface.

Parameter Name	Description
Source Port	Port from which the proxy should operate.
Source Path	Path from which the proxy should operate.
Target IP address	IP to which the proxy redirect. Note: mandatory.
Target port	Port to which the proxy redirect. Note: mandatory.
TLS enable	Proxy over TLS.

18.11 HTTP redirect

> **Info**
>
> Application available since SBC release 4.6.

> **Info**
>
> Support dropped starting 5.5.

Setup an HTTP redirect pattern, using nginx rewrite directive: RewriteDoc.

The template (`/etc/frafos/templates/nginx/http_redirect.tmpl`) may be overloaded, as described in Command Line Reference.

If "TLS enable" is set, a TLS profile is required.

The application may be enabled on `CX` interface.

Parameter Name	Description
Port	Port from which the redirect should operate.
Path	Path from which the redirect should operate. Path is a regex to which we prefix ^ (start of line).
Target URL	URL to where be redirected. Note: mandatory.
TLS enable	Redirect over TLS.

18.12 Prometheus Pull Service

Info

Application available since SBC release 5.2.

Info

Starting 5.5 Prometheus statistics are available via Unified SBC management service application and Prometheus Pull Service application is deprecated.

Enable the prometheus pull service application on SBC node, allowing external prometheus scrapers to query the pull service to get statistics on the SBC.

The application may only be enabled on `CX` interfaces.

Parameter Name	Description
Port	The http(s) port of prometheus pull service.
Path	The url path part which to serve the statistics on.
TLS enabled	Use TLS on the pull service. Plain http will not be allowed. This can follow the configuration of the TLS profile (i.e. auth with trusted clients).
HTTP Auth. Username	Whether or not to use HTTP basic authentication on the pull service.
HTTP Auth. Password	Whether or not to use HTTP basic authentication on the pull service.
Threads	Number of threads to use while serving the requests.
Update interval	Interval in milliseconds to update the served statistics.