# FRAFOS ABC SBC Reference Book

***Release 5.5.2***

**FRAFOS GmbH**

**Jun 04, 2025**

# 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*

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*
- *Enable transparent dialog IDs*
- *Call transfer handling*
- *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)*
- *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 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.

**Note:** 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.

**Note:** 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 host 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 hostname 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 hostname 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.

**Note:** 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) or with the tool **`sbc-calc-ha1`** 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.

**Note:** '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.

**Note:** 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.

**Note:** "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.

**Parameters**

Leg: A or B parameter name parameter value

### 1.1.41 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

### 1.1.42 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

### 1.1.43 Forward Contact-HF parameters

Forward all Contact header field parameters “as is” downstream.

### 1.1.44 Forward Contact-URI parameters

Forward all Contact URI parameters "as is" downstream.

Available since: 4.6.

### 1.1.45 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>
...
```

### 1.1.46 Translate Reply Code

Translate SIP reply codes to other value.

**Parameters**

matching reply code

new reply code

new reason phrase

### 1.1.47 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

### 1.1.48 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).

### 1.1.49 Forward Via-HFs

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

### 1.1.50 Diversion to History-Info

Converts SIP Diversion header-field into History-Info.

### 1.1.51 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.

### 1.1.52 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.

### 1.1.53 Handle INVITE with Replaces header

Activates internal processing of INVITE with Replaces header.

### 1.1.54 Map Replaces header

Activates mapping of dialog identifiers in INVITE with Replaces.

### 1.1.55 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

### 1.1.56 Set Content Type whitelist/blacklist

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

**Parameters**

comma-separated list of content types

### 1.1.57 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

### 1.1.58 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

### 1.1.59 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 Media whitelist

Permit only listed media types.

**Parameters**

media list

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

### 1.2.7 Set Media blacklist

Remove listed media types.

**Parameters**

media list

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

### 1.2.8 Drop early media

Drop early media (audio only).

### 1.2.9 Drop SDP from 1xx replies

Drop SDP from listed 1xx replies.

**Parameters**

list of affected reply codes

### 1.2.10 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.

### 1.2.11 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.

### 1.2.12 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.

### 1.2.13 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.

### 1.2.14 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.

### 1.2.15 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.

### 1.2.16 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.

### 1.2.17 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.

### 1.2.18 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.

### 1.2.19 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

**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.

**Note:** 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 *alternativen1n2n3n4n* 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 %}*.

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.

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 call agent use the same SSRC value per per stream. The SSRC is generated randomly for each call and is derived using the SDP media index of the stream.

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 ABC 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.

**Important:** 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:

- *AWS Parameters*
- *Backup Parameters*
- *CDR Parameters*
- *Event Parameters*
- *Eventbeat Parameters*
- *Firewall Parameters*
- *LDAP Parameters*
- *Lawful Interception Parameters*
- *Login*
- *Low-level Parameters*
- *Miscellaneous Parameters*
- *Meet-Me web conference Parameters*
- *System Monitoring Parameters*
- *PCAP Parameters*
- *SEMS Parameters*
- *SIPREC Parameters*
- *SIP Parameters*
- *SRTP Parameters*

- *Syslog Parameters*
- *Signaling SSL*
- *RTP handling Parameters*

## 2.1 AWS Parameters

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

At this moment they are used for performing initial AWS config when using HA under AWS.

Note that anyone in possession of an AWS IAM User Access key may impersonate the key's owner. It therefore makes sense to create a user with limited permissions and access AWS from the ABC SBC under this user's identity. Read the following link to learn more about IAM user identities: https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_access-keys.html

Table 1: AWS Parameters

| Parameter Name | Description |
|---|---|
| Region for AWS requests | AWS Region.<br>Available since: 4.3 |
| AWS access KEY ID | Key ID of an AWS user who was permission for the AWS service<br>Available since: 4.3 |
| AWS secret access KEY | The secret associated with the AWS user's key id. Note that the secret is only revealed when they key is created. When forgotten, the key must be created newly. When leaked, anyone in possession of the key may impersonate the user.<br>Available since: 4.3 |

## 2.2 Backup Parameters

These parameters set ABC SBC daily backups. See also more in Sec-Backup.

Table 2: Backup Parameters

| Parameter Name | Description |
|---|---|
| Equivalent settings as for CCM | If enabled, the settings on this Backup tab will not be applied on Sbc nodes, but the same settings as configured for CCM node (under CCM / CCM Config / Backup page) will be applied to Sbc nodes instead. |
| Create daily Sbc configuration backups | If enabled, daily snapshot of ABC SBC configuration will be created into backup gzipped tarball file. |
| Include provisioned tables in daily backups | If enabled, the daily backup will include also content of whole provisioned tables. |
| Number of days to keep backups | Sets the retention period for backup files. All files named sbc-backup-* in the backup directory older than specified number of days will be deleted on every daily backup run. Use 0 to disable automatic deletion of old backup files. |
| Destination directory for backups | Specifies the destination directory for the daily backup files. Default is "/data/backups" directory. |
| Full path to extra files or dirs to include in backup | Extra custom files or directories to be included in backup, using full paths, more fields separated by comma. A * wildcard can be used. The path must not contain comma character. |

## 2.3 CDR Parameters

These parameters allow to define how and where CDRs are stored. See also more in Sec-CDR.

Table 3: CDR Parameters

| Parameter Name | Description |
|---|---|
| Enable CDRs | Enable writting CDRs. |
| Number of CDR files to keep | CDR Retention policy. The ABC SBC produces CDRs for all completed calls in CSV form. Sets number of CDR files to keep. |
| Directory for exported CDR files: | Directory in filesystem where the CSV CDRs are stored. |
| CDR files rotation frequency (daily,weekly, monthly) | Sets the frequency of CDR files rotation. Use "daily", "weekly" or "monthly". The number of rotated files to keep before deletion is set using the "Number of CDR files to keep" |
| Enable new version of CDRs (CDR-NG) | Enables new version of CDRs, called CDR-NG. This feature is in experimental state in ABC SBC 4.5 release. |

## 2.4 Event Parameters

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

Table 4: Event Parameters

| Parameter Name | Description |
|---|---|
| Number of days to keep old traffic log files | Local retention policy. Particularly useful when no ABC Monitor is attached to the ABC SBC. Must be shorter than the retention policy at ABC Monitor – otherwise the ABC SBC may keep copying files that already expired at ABC Monitor. |
| ABC Monitor address | IP address or DNS name of ABC Monitor.<br>Empty if no ABC Monitor is attached to the ABC SBC. |
| Secondary ABC Monitor address | IP address or DNS name of secondary ABC Monitor. Empty if no secondary ABC Monitor is attached to the ABC SBC. |
| Replicate traffic logs to ABC Monitor | Allows to push collected PCAPs to a Monitor server using the rsync protocol. The files are deleted from Sbc after transfer. |
| Replicate traffic logs to secondary ABC Monitor | Allows to push collected PCAPs to a Monitor server using the rsync protocol. The files are deleted from Sbc after transfer. |
| Replicate recordings to ABC Monitor | Allows to push recorded audio files (see Section recording) to a Monitor server using the rsync protocol. The files are deleted from Sbc after transfer. |
| Replicate recordings to secondary ABC Monitor | Allows to push recorded audio files (see Section recording) to a Monitor server using the rsync protocol. The files are deleted from Sbc after transfer. |
| Replication rsync password | rsync password to be used for replicating traffic logs and recorded audio. |
| Replication rsync password for secondary ABC Monitor | rsync password to be used for replicating traffic logs and recorded audio. |

continues on next page

Table 4 – continued from previous page

| | |
|---|---|
| Use secure TLS connection to ABC Monitor | If enabled, events, traffic log and recording files will be pushed to ABC Monitor over TLS secured connection. It is highly recommended to install trusted certificate for this on ABC Monitor end instead of default self-signed. On Sbc side, the TLS profile of IMI interface is used. |
| Number of hours to keep old recordings (0 to not delete) | Retention policy for recored WAV files. |
| Generate an event if a SIP transaction reaches the defined number of retransmissions | Allows to monitor failing incoming transactions and detect SIP UACs with connectivity issues. The events are of type "notice" and appear in ABC Monitor's Transport Dashboard. Use with care, a too low number will result in dramatic increase of events. If used, recommended value is 4. |
| Maximum number of events buffered in local Redis | Retention policy for locally buffered events |
| List of call variables added into events | Contain list of call variables that are added into call events.<br>The list shall contain comma separated pairs: *<var_name>:<flag>* where *<var_name>* is name of call variable and *<flag>* is 0 or 1 specifying whether the value of call variable can be overwritten. User may use the wildcard (*) character to denote ALL events. |
| Generate an event on UDP receive buffer errors | If enabled, alert event will be generated if UDP receive buffer errors are detected on system network interface. |
| Generate an event on UDP send buffer errors | If enabled, alert event will be generated if UDP send buffer errors are detected on system network interface. |
| Generate an event on UDP packet receive errors | If enabled, alert event will be generated if UDP packet receive errors are detected on system network interface. |
| Generate an event on IP incoming packet receive errors | If enabled, alert event will be generated if IP incoming packet receive errors are detected on system network interface. |
| Generate an event on outgoing packets dropped errors | If enabled, alert event will be generated if outgoing packets dropped errors are detected on system network interface. |
| Alarm when number of calls reaches % of the license. | sems will yield a warning message once the number of session reached X% of the license limit. A downstream message is also yield (info level), once the number of session go below X%.<br>Default: 75. |
| Privacy monitor mode | sems will not send call-attempt, call-start and call-end events to monitor. Can be overridden via "Disable privacy monitor mode" action.<br>Default: off |
| Destination monitor event interval (sec) | Interval at which destination monitor events are generated. Value is in seconds.<br>Default: 300 (5min) |
| Threshold of number of events buffered on Sbc to set warning | If there are more events waiting in redis queue on Sbc side than the limit set here, the node status will be set to warning on System monitoring page.<br>Default: 500 |
| Enable events redis disk persistence | Enables events redis disk persistence, using /data/redis directory. Use with caution, there has to be enough disk space. |

continues on next page

Table 4 – continued from previous page

| | |
|---|---|
| Periodic RTP Statistics | Enables sending of periodic RTP statistics per call-leg and at 10 second intervals.<br>Available since: 5.4 |

## 2.5 Eventbeat Parameters

These parameters allow to tweaks and debug the event communications between an ABC SBC node and an ABC Monitor one. Some statistics may be generated and exposed on the application interface TCP port (*:4247* and *:4248*).

Table 5: Eventbeat Parameters

| Parameter Name | Description |
|---|---|
| Event batching size | Maximum number of event sent at once to the monitor. |
| Enable enventbeat statistic reporting | Expose on the TCP port (*:4247* for *sbc-eventbeat-1* and *:4248* for *sbc-eventbeat-2* some live metrics about events processing. |
| Interval between each statistic | Interval on which a single statistic entity was recorded. |
| How many statistic entries per payload | How many statistics entities should be returned in a single payload.<br>Ex: To have statistic about the last minutes, per packets of 5 seconds, set the following :<br>• *Interval between each stat* to 5<br>• set *How many entries per payload* to 12 |

## 2.6 Firewall Parameters

Table 6: Firewall Parameters

| Parameter Name | Description |
|---|---|
| Enable Sbc firewall | If enabled, the firewall chains will be filled with Sbc firewall rules. If deployed on container or system not supporting nftables or nfsets, this option has to be disabled, otherwise a Sbc node error will be reported in System status.<br>Note: on Sbc < 5.4 the firewall uses iptables and ipsets.<br>Available since: 5.0 |
| Reject packets instead of dropping | If disabled, the firewall silently drops packets not allowed. If enabled, packets will be rejected and icmp admin-prohibited message sent back instead.<br>Note: on Sbc < 5.4 the firewall uses reject, starting with 5.4 it uses drop by default.<br>Available since: 5.4 |
| Do not accept any icmp packets | If enabled, incoming icmp packets are dropped and not answered. Use with care, as blocking icmp may not be a good idea.<br>Available since: 5.5 |

continues on next page

Table 6 – continued from previous page

| | |
|---|---|
| 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 put on blacklist, silently dropping all packets from it.<br>Note that the individual reasons for blacklisting have to be also enabled in CA settings or in the Drop or Log message / Event for replies actions parameter. See Section Sec-Abuse for more details.<br>Available since: 4.3 |
| Signaling failures blacklist: IP address start score before any offense | Sets the score used as a starting value before any offense has been registered. This start value will be decreased each time until it reaches 0 or less, which finally leads to the blacklisting of the incriminated IP address. See Section Sec-Abuse for more details.<br>Available since: 4.3 |
| 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. This allows the score to recover slightly over time and thus can be understood as a bonus for good behavior. See Section Sec-Abuse for more details.<br>Available since: 4.3 |
| 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 from a certain IP address has been seen, that IP address is removed from the scoring table. Should a new offense be registered from a deleted IP address, the start score will be used. This allows for keeping the scoring table at a reasonable size. See Section Sec-Abuse for more details.<br>Available since: 4.3 |
| Time in seconds to blacklist IP addr for signaling failures | Sets the time how long the IP address will be held on blacklist, before removing it from blacklist automatically (for drop, failed auth, limit, sanity). See Section Sec-Abuse for more details.<br>Available since: 4.3 |
| 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 Sec-greylisting for more details.<br>Available since: 4.3 |
| 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 ABC SBC signaling interface for this to work. All traffic from the IP addresses on blacklist will be silently dropped. See Section Sec-greylisting for more details.<br>Available since: 4.3 |
| 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 Sec-greylisting for more details.<br>Available since: 4.3 |

continues on next page

Table 6 – continued from previous page

| | |
|---|---|
| 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 Sec-greylisting for more details.<br>Available since: 4.3 |
| Greylist: additional ports or port ranges (a:b) to check in addition to signaling ports, space separated | Sets additional ports to ports defined on ABC 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 Sec-greylisting for more details.<br>Available since: 4.3 |
| Blacklist: Log blacklisted IP addresses to syslog | Log blacklisted IP addresses to syslog. Entries are logged in the following file: '/var/log/frafos/sems-blacklist.log'<br>Available since: 4.3 |
| Greylist: Log greylisted IP addresses to syslog | Log greylisted IP addresses to syslog Entries are logged in the following file: '/var/log/frafos/sems-greylist.log'<br>Available since: 4.3 |
| 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.<br>Available since: 4.3 |

## 2.7 LDAP Parameters

ABC SBC allow authentication against an LDAP server. The authentication is done using the *nslcd* and *pam* packages. Once configured, users may then access an ABC SBC container, via *ssh*, using their UID and password.

Note: LDAP authentication is only available up to 5.4 SBC.

Table 7: LDAP Parameters

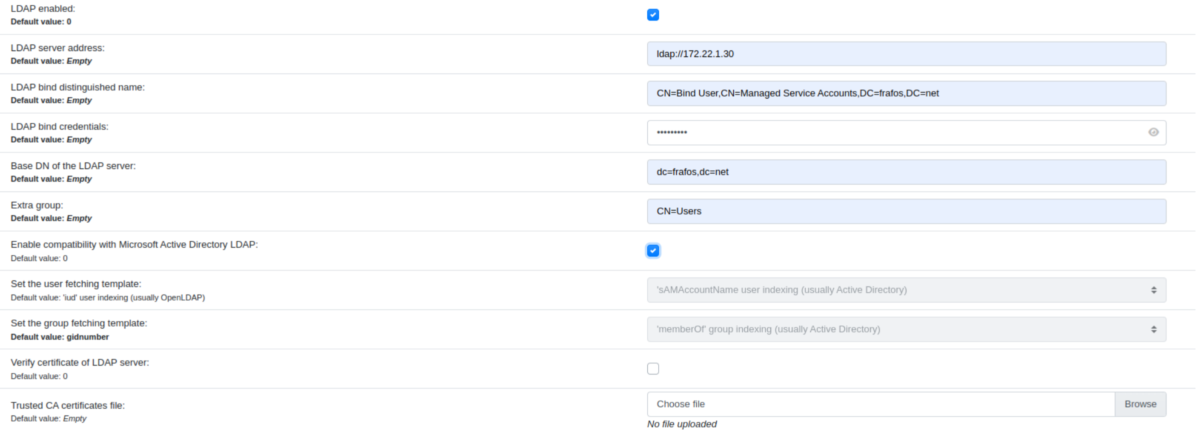
| **Parameter Name** | **Description** |
|---|---|
| LDAP auth enabled | Enable LDAP authentication.<br>Available up to: 5.4 |
| LDAP server address | LDAP host on which the LDAP service can be reached (ldap://IP:PORT or ldap://IP or ldap://my.domain)<br>Available up to: 5.4 |
| LDAP distinguished name / admin user DN | Specifies the distinguished name used to bind to the LDAP server for lookups.<br>Available up to: 5.4 |
| LDAP credentials / admin user PW | Specifies the LDAP credentials used to bind.<br>Available up to: 5.4 |
| base DN such as 'dc=example,dc=org' | Default search DN of the LDAP.<br>Ex: For "cn=admin,dc=example,dc=org", base DN is "dc=example,dc=org"<br>Available up to: 5.4 |

continues on next page

Table 7 – continued from previous page

| | |
|---|---|
| extra group such as 'ou=People' like in "uid=john,ou=People ,dc=example,dc=org" | So user only need to register their name (aka "uid") please pass any extra bind dn via this parameters.<br>Ex: user (like *john*) exist in the form, "uid=john,ou=People,dc=example,dc=org", so we set the following to "ou=People". GUI will then concatenate in the form uid=[user value][extra_group][base_dn] to auth the user against the ldap server.<br>Note that to complete a user login, the ldap user must **also** be member of a group matching one of the GUI groups supporting login. This group must be a primary group of that user.<br>Available up to: 5.4 |
| Enable Active compatibility with Microsoft Active Directory LDAP | Connect to an Active Directory LDAP server.<br>Available up to: 5.4 |
| Enable Active compatibility with IBM LDAP | Connect to an IBM LDAP server.<br>**limitation**: currently, CCM' matching group can only be done against a group name, instead of a full group cn. **example**: *group1* is valid, while *cn=group1,dc=frafos,dc=org* will fail.<br>Available up to: 5.4 |
| Check SSL/TLS peer certificate | Enable the check of client certificates. Please note that an Active Directory LDAP **needs** the certs to be configured in`/etc/openldap/certs`<br>Available up to: 5.4 |
| CA certificate for the LDAP | List of certificates to which the client's one are check. The certificate must be in PEM format.<br>Available up to: 5.4 |

Example of an ldap configuration:



| | |
|---|---|
| LDAP enabled:<br>**Default value: 0** | ☑ |
| LDAP server address:<br>**Default value: *Empty*** | ldap://172.22.1.30 |
| LDAP bind distinguished name:<br>**Default value: *Empty*** | CN=Bind User,CN=Managed Service Accounts,DC=frafos,DC=net |
| LDAP bind credentials:<br>**Default value: *Empty*** | ••••••••• |
| Base DN of the LDAP server:<br>**Default value: *Empty*** | dc=frafos,dc=net |
| Extra group:<br>**Default value: *Empty*** | CN=Users |
| Enable compatibility with Microsoft Active Directory LDAP:<br>Default value: 0 | ☑ |
| Set the user fetching template:<br>Default value: 'iud' user indexing (usually OpenLDAP) | 'sAMAccountName user indexing (usually Active Directory) |
| Set the group fetching template:<br>**Default value: gidnumber** | 'memberOf' group indexing (usually Active Directory) |
| Verify certificate of LDAP server:<br>Default value: 0 | ☐ |
| Trusted CA certificates file:<br>Default value: *Empty* | Choose file Browse<br>*No file uploaded* |

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

The image come in with 2 users (+ admin) :

| User | dn | pwd |
|---|---|---|
| john | *uid=john,ou=People, dc=example,dc=org* | *johnldap* |
| jane | *uid=jane,ou=People, dc=example,dc=org* | *janeldap* |

On some setup, it **may** be **requested** to append the user name to the "List of sshd allowed users" parameter (Miscellaneous Parameters).

You can then login with the credential *john* and the password *johnldap*:

```
$ grep 'AllowUsers' /etc/ssh/sshd_config
AllowUsers root john
$ ssh jane@127.0.0.1
jane@127.0.0.1's password: janeldap
Permission denied, please try again.
^C
$
$ ssh john@127.0.0.1
john@172.42.0.1's password: johnldap
Last login: Thu Jul 21 11:52:37 2022 from 192.168.1.21
john@yopyop:/home/jone/$
```

## 2.8 Lawful Interception Parameters

This is configuration of Lawful Interception.

Table 8: Lawful Interception Parameters

| Parameter Name | Description |
|---|---|
| Lawful Interception enabled | Enable the feature generally. Note that it has to be used also under corresponding action to take effect. |
| Operator ID | Set the Operator ID value. |
| Delivery Country Code (DCC) | Set the Delivery Country Code (DCC) value. |
| Network Element Identifier | The Network Identifier (NID) consists of the operator identifier and, optionally, the network element. |
| Interception Point ID | Set the Interception Point ID value. |

## 2.9 Login

Parameters related to login/logout.

Table 9: Login Parameters

| Parameter Name | Description |
|---|---|
| Time for terminal session automatic logout if idle, in seconds | Sets the time in seconds after which idle terminal session to ABC SBC will be automatically closed. Default value is 600 sec. Use 0 to disable. |

## 2.10 Low-level Parameters

These settings have effect only after reboot of the server. Additional information can be found in the Section Sec-Hardawre-conf.

Caution: changing these parameters may dramatically change system behavior. Their effect largely depends on used equipment.

Table 10: Low-level Parameters

| Parameter Name | Description |
|---|---|

continues on next page

Table 10 – continued from previous page

| | |
|---|---|
| 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.<br>Available since: 4.3 |
| Interfaces where to set ethtool options | Network interfaces where to apply the following coalesce and ringbuffer ethtool options. Separated by spaces.<br>Available since: 4.3 |
| Coalesce ethtool options | Ethernet adapter coalescing options, syntax of ethtool. Applied on interfaces listed in “Interfaces where to set ethtool options”. This option allows to fine-tune a trade-off between less-CPU-intensive and more-real-time packet processing in kernel. The tuning outcome is specific to used network card.<br>Available since: 4.3 |
| Ringbuffer ethtool options | Ethernet adapter rx/tx ring parameters, syntax of ethtool. Applied on interfaces listed 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.<br>Available since: 4.3 |
| 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.<br>Available since: 4.3 |
| Run db check on boot | If enabled, run “mysqlcheck” command during boot process. This option allows a safe recovery from an unexpected shutdown and is therefore by default turned on. The check may slowdown machine startup. |
| Clean tmp files on boot | If enabled, clean-up system directory for temporary files. |
| Sems memory limit in % from total memory | Limit Sems process memory maximum usage. Set to 0 for no limit. |
| Provisioned tables redis disk persistence time interval (in seconds) | Sets the time interval after which provisioned tables data on Sbc backup node will be saved from in-memory redis database to disk to allow persistence for reboot. May be tuned according to provisioned tables data size. The data is saved if there were more record changes than set via the following setting for minimum number of records. Default is 600 seconds. |
| Provisioned tables redis disk persistence number of records to trigger save | Set minimum number of provisioned tables record 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 is 1 record. |
| 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. Can be used only if 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. |
| Session processor threads | These threads process the SIP signaling of the sessions. They also process the B (routing) and C (outbound) rules of the ABC rule set. created in a thread pool among which all SIP sessions are distributed. Usually we recommend to set this to the number of usable hardware threads on the CPU multiplied by two, but to no less than 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. |
| Media processor threads | These threads process RTP media for transcoding and media applications like conferencing and 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 two times the usable CPU hardware threads, otherwise it is recommended to leave them to the default (16) or even less. |

continues on next page

Table 10 – continued from previous page

| | |
|---|---|
| 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. |
| RTP receiver threads | These threads receive RTP packets and relay them. They also decrypt SRTP packets if 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 two to four times the usable CPU hardware threads. |
| Call restore threads (HA) | This is a thread pool that is only used when doing the call restore after failover. It is recommended to set it to the number of usable CPU hardware threads. |
| Out-of-dialog requests threads | These threads handle REGISTER, SUBSCRIBE/NOTIFY and MESSAGE requests. If a lot of registrations are handled, or a lot of subscriptions, then it is recommended to set this to a higher number. |
| HA interval to send adverts to peer, in seconds | Set the HA interval for keepalived daemon to send adverts to it's HA peer. Decimal number allowed. Lower values shorten detection time for HA switchover, but be careful when setting low value, as too low interval may bring stability issues. It is recommended to keep at default value for typical setups. |
| HA priority | Set the HA VRRP priority. Possible range is 1-255, where 255 has special meaning of address owner. It is recommended to keep at default value for typical setups. |
| HA - use unicast instead of multicast for vrrp adverts | If enabled, the HA vrrp protocol will use IP unicast adverts instead of default IP multicast. |
| HA - use also signaling interfaces for vrrp adverts | If enabled, the HA vrrp protocol adverts will be sent not only on IMI or custom interface where the HA app is used, but additionally also on signaling interfaces. Note that for this to work, the system interface corresponding to signaling interface has to use also some non VIP IP address. Also the vrrp adverts using IP unicast must not be enabled (see option above). |
| HA - use macvlan interface for VIP | If enabled, a separate macvlan sub-interface will be created for the VIP address. This is usually not needed, but can help to achieve faster switchover under some circumstances.<br>Available since: 5.5 |

## 2.11 Miscellaneous Parameters

Table 11: Miscellaneous Parameters

| Parameter Name | Description |
|---|---|
| Permit root login using ssh | Sets if root is allowed to login to ABC SBC server using ssh. Use 'yes' to allow root login, or 'prohibit-password' to allow login but password and keyboard-interactive authentication disabled, or 'no' to disable. |
| List of sshd allowed users | List of users allowed to login via ssh, if the ssh app is enabled on Sbc interface. Use space to separate more entries. Use empty value to allow all users. |
| Enable ssh password authentication | Enables or disables PasswordAuthentication option in sshd config. Default is enabled. |
| Blacklist timeout for IP addresses from external sources | Timeout in seconds for the IP addresses blacklisted by RESTFul requests. |

continues on next page

Table 11 – continued from previous page

| | |
|---|---|
| Enable sending important syslog entries to ABC monitor | Enables or disables sending syslog entries of levels 'critical' up to 'emergency' as an alert to the ABC monitor. |
| Automatically add new nodes | If enabled, records for new nodes that pull config from configuration master will be automatically added. If disabled, the configuration master will refuse to provide configuration to nodes that are not already defined in Nodes configuration. |
| Session Management enable | Enables advanced load-balancing, see more details in the section :ref:Sec-adv-load-balancing |
| Failed system login lock unlock time | Time in seconds to keep system accounts locked after 3 failed login attempts. Default value is 600 seconds. |
| Geoip - account id for geoipupdate command | Used to pass account id for geoipupdate command, which is run periodically if the license is provided to retrieve geoip GeoLite 2 database. The license has to be created by user using his MaxMind account. |
| Geoip - license key for geoipupdate command | Used to pass license key for geoipupdate command, which is run periodically if the license is provided to retrieve geoip GeoLite 2 database. The license has to be created by user using his MaxMind account. |

## 2.12 Meet-Me web conference Parameters

Table 12: Meet-me conf Parameters

| Parameter Name | Description |
|---|---|
| Keep participant's name file for (hours) | Settings defining for how long files holding webconference participant's name will be kept on the FS (not subject to replications). |
| Echo the number of participant on event | If enable, the number of participant is echo'ed when a participant join or leave the conference room. Alternatively, one may press the star (*) key while in call to achieve the same. |
| Use room security pin value for the admin pin | If the 'Use security pin' and 'Use admin pin' options are enabled for a room, then the 'admin pin' value is set to the same as the 'security pin'. |
| Path to directory holding digits wav files | The files are used to echo numbers. File expected hold values like 'one', 'twenty', '(seven-)teen' etc … By default the SBC ships 2 flavors of that directory: */usr/lib/sems/audio/webconference/digits/* for English prompt, and */usr/lib/sems/audio/webconference/de/digits/* for Germans one. |
| List of provtables table to watch for expired generated room | Generated webconference name and PIN are persisted to the CCM provtables. The CCM's configured to attempt to remove expired room from the following listed provtable every day at 2am. |
| Generated rooms validity (days) | Number of days generated conference room are considered as 'open'. Once a room's closed, it's PIN's blocked for a fixed amount of time. |
| Keep expired generated rooms (days) | Number of days before closed generated conference room's PIN are unblocked. |

## 2.13 System Monitoring Parameters

These parameters allow to set up an email alarm if system resources are used excessively.

Not that this same email is used to setup the Let's encrypt auto certification.

Table 13: System Monitoring Parameters

| Parameter Name | Description |
|---|---|
| email for sending alerts | Email address to which important alerts like reports on excessive CPU usage are sent. Use empty value to disable sending the email alerts. This email address will be also used in case of let's encrypt auto certificate renew on TLS profile.<br>Available since: 4.3 |
| mailserver for sending alerts | Specifies address of SMTP server used as email relay. Note: when ABC SBC is running in container, mail relay on localhost is not available and external mail server has to be used.<br>Available since: 4.3 |
| SMTP mail server port | Set the SMTP mail server port.<br>Available since: 5.1 |
| Use secure connection to SMTP mailserver | Set if the SMTP connection to mailserver should be encrypted, and if yes if using TLS or STARTTLS.<br>Available since: 5.1 |
| SMTP mail server authentication | **Use 'off' to disable the authentication, or 'on' to enable**<br>it and choose auth type automatically.<br>Available since: 5.1 |
| Username for SMTP authentication | Set the username for SMTP authentication, if authentication is enabled.<br>Available since: 5.1 |
| Password for SMTP authentication | Set the password for SMTP authentication, if authentication is enabled.<br>Available since: 5.1 |
| from address for sending alerts | email address used for From in email alerts, system default is used if empty<br>Available since: 4.3 |
| 1min load threshold | CPU load threshold which if exceeded for one minute will raise an alarm. The load threshold values should be set correspondingly to system CPU cores number.<br>Available since: 4.3 Available up to: 5.4 |
| 5min load threshold | CPU load threshold which if exceeded for five minutes will raise an alarm (typically lower value than previous). The load threshold values should be set correspondingly to system CPU cores number.<br>Available since: 4.3 Available up to: 5.4 |
| cpu wait % threshold | threshold for % of CPU time in wait status to raise an alarm<br>Available since: 4.3 Available up to: 5.4 |
| memory usage % threshold | threshold of memory occupation in % which if exceeded will raise an alarm<br>Available since: 4.3 Available up to: 5.4 |
| disk usage % threshold | threshold of disk usage in % which if exceeded will raise an an alarm<br>Available since: 4.3 Available up to: 5.4 |
| send system monitoring data to ABC Monitor | if remote ABC monitor is used, send system monitoring data to it together with signaling events<br>Available since: 4.3 |
| send extended system info emails when over treshold | if enabled, email with more detailed system information will be sent when some monitoring threshold is reached<br>Available since: 4.3 |
| extended info emails frequency | limit frequency of sending the extended info emails, use value with min, hour or day suffix<br>Available since: 4.3 |
| Check status of system interfaces | If enabled, system network interfaces will be periodically checked and alert events created if errors are detected. Individual check types can be set using following options |

## 2.14 PCAP Parameters

These parameters allow to set up how the most recent SIP traffic is recorded on the system for sake of troubleshooting. The ABC SBC stores the SIP traffic in PCAP files of given size and deletes the least recent files. The PCAP files can be inspected in the administrative interface as shown in Section Sec-user-recent-traffic.

Table 14: PCAP Parameters

| Parameter Name | Description |
|---|---|
| File size in MB for one pcap file | maximum size of a PCAP file after which a new file is created |
| Number of pcap files to keep | PCAP retention policy. PCAP files are rotated and only the configured number of PCAP files is kept. The least recent files are deleted. Use 0 to disable storing SIP traffic completely, which is not recommended because of troubleshooting. Note: the pcap filenames are using extension ".pcapXX" where XX corresponds to the file number. If the number of files is modified, all existing traffic.pcap* files are deleted once the configuration change is activated. |

## 2.15 SEMS Parameters

These parameters determine the behavior of the ABC-SBC "engine", the SEMS signaling and media processor. The parameters are used primarily for troubleshooting and performance tuning and shall be therefore changed only when there is a good reason for doing so.

Table 15: SEMS Parameters

| Parameter Name | Description |
|---|---|
| Use raw sockets | Performance optimization techniques for sending RTP packets on Linux systems with slow UDP stack. |
| Default Destination Blacklist TTL | Defines how long are unavailable IP destinations maintained on a blacklist to which no SIP traffic is sent by default. For Call Agent, a specific value may be entered in the Call Agent parameters. See Sec-adaptive-blacklisting. |
| Persistent redis storage | If enabled, the calls and registrations state data that is stored in redis db, will be preserved during server reboot. |
| Interval in seconds for saving if persistent storage enabled, use 0 to disable | This sets the time interval in seconds, after which the calls and registrations data will be saved periodically if the Persistent redis storage is enabled. Use with caution, on big setups it can add additional load on the server. Use 0 to disable the periodical saving, which means the state will be saved only on restart done due to config activation, or container reboot. |
| Load q850_reason call control module | If enabled, the module for processing Q.850 reasons will be loaded. The cc_q850_reason.conf is empty by default and it can be used only if custom local template for this config file is provided (/data/local-templates/etc/sems/cc_q850_reason.conf). |

continues on next page

Table 15 – continued from previous page

| | |
|---|---|
| Mariadb timeout for "Read call variables" queries | Timeout (in seconds) of Mariadb queries done when reading call variables using the action or condition "Read call variables".<br>The main purpose of this parameter is to reduce problems caused by queries that may take too much time and block processing of other calls.<br>Please note that timeout of such Mariadb queries means that system is either overloaded or blocked and the root cause should be fixed instead of tuning the timeout value.<br>Negative value or 0 means that default timeout of the MySQL++ library will be used.<br>Default value: 5 |
| Websocket ping-pong interval in seconds | Interval in seconds to send keepalive ping-pong messages on Websocket signaling interfaces. Use 0 to disable. |
| Soft limit for out-of-dialog transactions | Number of active server transactions that, if passed, will trigger an alert event. This limit will only be taken into consideration when creating a server transaction which is not related in any way to an existing dialog.<br>Use 0 to disable that feature.<br>See section Sec-trans_limits for more details. |
| Hard limit for out-of-dialog transactions | Limit for the number of active server transactions, which will be enforced when creating a new server transaction not related to an existing dialog. The limit is enforced by replying to new requests with "503 Overloaded". Additionally, a corresponding monitoring event will be created.<br>Use 0 to disable that feature.<br>See section Sec-trans_limits for more details. |
| Event throttling for soft/hard OOD limit | Throttle the events generated by the hard & soft limit for out-of-dialog transactions to no more than one of each type (soft / hard) per configured time lapse in seconds.<br>Use 0 to disable that feature.<br>See section Sec-trans_limits for more details. |
| Soft limit for in-dialog transactions | Number of active server transactions that, if passed, will trigger an alert event. This limit will only be taken into consideration when creating a server transaction related to an existing dialog.<br>Use 0 to disable that feature.<br>See section Sec-trans_limits for more details. |
| Hard limit for in-dialog transactions | Limit for the number of active server transactions, which will be enforced when creating a new server transaction related to an existing dialog. The limit is enforced by replying to new requests with "503 Overloaded". Additionally, a corresponding monitoring event will be created.<br>Use 0 to disable that feature.<br>See section Sec-trans_limits for more details. |
| Event throttling for soft/hard DLG limit | Throttle the events generated by the hard & soft limit for in-dialog transactions to no more than one of each type (soft / hard) per configured time lapse in seconds.<br>Use 0 to disable that feature.<br>See section Sec-trans_limits for more details. |

continues on next page

Table 15 – continued from previous page

| | |
|---|---|
| Loop detection secret | This parameter is used to create a special branch tag. When we receive a new request that contains our prepared tag in the first Via-HF, we refuse the request with 482 Loop detected. Use empty value to disable, “auto” for automatic secret string, or provide a string. Default is “auto”. |
| Strict checking of the user part of a URI to only allow chars as per RFC3261 | When the strict checking is enabled, the user part of a URI is only allowed to contain the chars as per RFC3261 (see ABNF rules). When disabled, ABC SBC does everything to let the most through, as long as it does not prevent it from parsing URIs correctly. Enabled by default. |
| TCP connection idle timeout in milliseconds | Sets TCP connection timeout if idle, on signaling interfaces. Use value in milliseconds, or 0 to disable. |
| Delay after startup to ignore limits | Delay in seconds to ignore CAPS and other limits after start of ABC SBC signaling application. |
| RESTful interface - verify https peer | If enabled, the validity of https certificate of peer will be verified on RESTful interface queries. Enabled by default. |
| REST Custom CA file | If provided, SEMS’s rest module will use the provided custom CA for every outgoing https request.<br>ABC SBC will handle on it own the adding of the ca to the nodes trusted chain. Process:<br>• CA copied to */etc/pki/ca-trust/source/anchors/*<br>• run *update-ca-trust* |
| User-agent string | If provided, the input here is used to set the User-Agent on SIP messages. |
| TCP send timeout for signaling interfaces | Set TCP connection timeout in milliseconds for signaling interfaces (see TCP_USER_TIMEOUT in tcp(7) man page). |
| Unprocessed events limit | If the REDIS server is offline, DB writes are queued internally. If REDIS is offline for a long time, the internal write queue can grow, using up a lot of memory. This parameter limits the write queue size; if the write queue has reached this size, further writes are ignored. Set to 0 to disable the limit. |
| Unprocessed events limit warning threshold | If the write queue grows over this threshold, SEMS warns by generating WARN level syslog messages. Set to 0 to disable. |
| Log every monitored destination state | Every state of a monitored destination (changed or not) will generate a log message on the INFO level. |
| Terminate calls on Sbc shutdown or restart | If enabled, Sbc will try to terminate calls on the container shutdown or restart.<br>Note: if using HA, it may be better to not terminate calls on Sbc shutdown or restart, it is recommended to set this option to disabled in that case. |
| DNS Resolver Timeout | Timeout in milliseconds before giving up waiting for a response to a DNS query. Default is 100. |
| DNS Cache Renew Period | Duration in seconds. It is used to early-refresh the entry in the cache if it would be expired after this duration. |
| DNS Cache Grace Period | Duration in seconds to wait before discarding an item from the cache after it’s expired. |

## 2.16 SIPREC Parameters

Table 16: SIPREC Parameters

| Parameter Name | Description |
|---|---|
| SIPREC outbound interface | Use Sbc interface name to force outbound interface for SIP recording. Leave empty by default. |
| SIPREC media interface | Sbc interface to be used for sending media towards SIPREC server. Empty by default. |
| SIPREC SIP timer A (ms) | SIP timer A used towards SIPREC server. Default value 500ms. |
| SIPREC SIP timer B (ms) | SIP timer B used towards SIPREC server. Default value 32s. |
| SIPREC SIP timer C (ms) | SIP timer C used towards SIPREC server. Default value 180s. |
| SIPREC SIP timer F (ms) | SIP timer F used towards SIPREC server. Default value 32s. |
| SIPREC SIP timer L (ms) | Timer L used towards SIPREC server. Default value 32s. |
| SIPREC SIP timer M (ms) | Timer M used towards SIPREC server. Default value 8s. |
| SIPREC SIP timer T2 (ms) | SIP timer T2 used towards SIPREC server. Default value 4s. |

## 2.17 SIP Parameters

These parameters set SIP timers, as defined in RFC 3261. All values are in ms.

Extra parameters are available, see the following table:

Table 17: SIP Parameters

| Parameter Name | Description |
|---|---|
| Add Q850 header to timer expiration's CANCEL. | If enable, then a Q850 header is added to the CANCEL generated by a timer expiration. Currently, only time C is supported. |
| Terminate dialog upon failure replies for in-dialog OPTIONS | Terminate dialog if in-dialog OPTIONS request fails with reply that should cause dialog termination.<br>Reply codes that should terminate the dialog according to RFC 5057 are: 404, 410, 416, 482, 483, 484, 485, 502, 604.<br>Additionally ABC SBC handles following replies the same way as those listed above: 408, 480.<br>Affects only INVITE based dialogs (i.e. calls).<br>The purpose of this option is to cope with interoperability issues caused by badly implemented SIP user agents that can't handle in-dialog OPTIONS correctly.<br>Default value: on (terminate the dialog) |
| Remove filtered m-lines | Remove media lines filtered out by media whitelist/blacklist. These lines are left in SDP but marked as inactive if not enabled.<br>This option is applied globally on all calls with active media whitelist or blacklist (see mediatypefiltering).<br>The purpose of this option is to cope with interoperability issues caused by badly implemented SIP user agents that can't handle inactive media streams correctly.<br>Default value: off (i.e. mark media lines as inactive) |

continues on next page

Table 17 – continued from previous page

| | |
|---|---|
| Try filling missing rtpmap in disabled media | On SDP answers with disabled media, SBC will try to fill in missing a=rtpmap and a=fmt information by copying them from the offer in case they are missing.<br>Default value: off<br>Available since: 5.3 |
| Filter forced transports | Remove media lines that do not match outbound transport forced by Force RTP/SRTP action (see rtpandsrtp). These lines are left in SDP but converted to the required transport if not enabled.<br>For example:<br>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).<br>SRTP is forced in outbound rules on ABC SBC.<br>If Filter forced transports option is “off” ABC SBC forwards SDP with two audio streams to the callee both of them over SRTP.<br>If this option is “on” ABC SBC forwards SDP with just one audio stream over SRTP to the callee.<br>This option is applied globally on all calls using Force RTP/SRTP action.<br>The purpose of this option is to cope with interoperability issues caused by user agents that can’t handle multiple media streams of the same type.<br>Default value: off (i.e. convert the media lines to the forced transport) |
| Call transfers using late offer-answer | Use offer-less INVITE when generating new call leg during call transfer (unattended call transfer or call transfer replacing non-local call).<br>It is probably the only reliable way that should work. Unfortunately too many SIP UAs do not implement late offer-answer correctly.<br>Default value: off |
| Predefined payloads for call transfers | Coma separated list of codecs to be added into SDP of INVITE generated during call transfer (unattended call transfer or call transfer replacing non-local call).<br>If no codecs are listed, only codecs used within the call are used what can cause troubles if the destination doesn’t support these.<br>Only simple codecs can be used (no parameters can be specified).<br>For example: PCMU,PCMA<br>Default value: empty |

continues on next page

Table 17 – continued from previous page

| | |
|---|---|
| Force outbound interface | If enabled, UDP packets sent will be forced to use the system interface attached to the outbound call agent.<br>Please note that this option relies on operating system capabilities that have heavy limitations.<br>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.<br>Manually configured source IP based policy routing is the preferred method.<br>Default value: off |

## 2.18 SRTP Parameters

These parameters define the security handshake of Secure RTP. SRTP is always used for WebRTC and is used with some encryption-enabled SIP devices.

Table 18: SRTP Parameters

| Parameter Name | Description |
|---|---|
| DTLS certificate file | Certificate file. Optional. Keep empty for self-signed certificate. That's the recommended configuration: other certificates may cause DTLS packets to become too large and consequently fail to traverse NATs due to IP fragmentation. |
| DTLS private key file | Private key file. Optional. |
| DTLS handshake timeout (ms) | Duration in milliseconds for handshake to be done before terminating the call. 0 disables it. |
| SRTP crypto-suite AES_CM_128_HMAC_SHA1 | Enables / disables the corresponding crypto suite. It should be left enabled unless required otherwise for interoperability. |
| SRTP crypto-suite AES_CM_128_HMAC_SHA1_80 | Enables / disables the corresponding crypto suite. It should be left enabled unless required otherwise for interoperability. |
| SRTP crypto-suite AES_256_CM_HMAC_SHA1_80 (SDES only) | Enables / disables the corresponding crypto suite. It should be left enabled unless required otherwise for interoperability. |
| SRTP crypto-suite AEAD_AES_256_GCM | Enables / disables the corresponding crypto suite. It should be left enabled unless required otherwise for interoperability (available since 5.2). |
| SRTP crypto-suite AEAD_AES_256_GCM_8 (SDES only) | Enables / disables the corresponding crypto suite. It should be left enabled unless required otherwise for interoperability (available since 5.2). |
| SRTP crypto-suite AEAD_AES_128_GCM | Enables / disables the corresponding crypto suite. It should be left enabled unless required otherwise for interoperability (available since 5.2). |
| SRTP crypto-suite AEAD_AES_128_GCM_8 (SDES only) | Enables / disables the corresponding crypto suite. It should be left enabled unless required otherwise for interoperability (available since 5.2). |
| SRTP crypto-suite preference order | Comma separated list of crypto suites. I.e. 'AEAD_AES_256_GCM , AEAD_AES_128_GCM'. Suites offered by the remote endpoint will always take precedence. Suites that are supported but not listed in this list are appended at the end according to the default order (available since 5.2). |
| DTLS legacy server connect | Enabled / disables supporting connection to legacy servers that do not support secure renegotiation extension. The default behavior is to terminate the DTLS handshake if this is not enabled. For detailed information, refer to RFC5746 sections 3.4 & 4.1. WARNING: Enabling this degrades security. This should only be used in cases where the DTLS servers the SBC connects to have renegotiation disabled!<br>Available since: 5.4 |

## 2.19 Syslog Parameters

These parameters allow to fine-tune behavior of syslog daemon. This is primarily useful when the syslogs are configured to be sent to an external system.

Table 19: Syslog Parameters

| **Parameter Name** | **Description** |
|---|---|
| Log level | This option changes the SEMS syslog globally. See the Section *Reference of Log Level Parameters* for a full list of options. |
| Syslog facility | Name of syslog facility to use for logs from the main SBC processes. Possible values are 'daemon', 'user', 'local0', 'local1' … 'local7'. |
| Enable remote syslog servers | If turned on, syslog messages will be sent to an external syslog host(s) additionally to the local filesystem. |
| Remote syslog server address | Address of the external syslog server. |
| Remote syslog server port | Port number on which the external syslog server listens. |
| Remote syslog transport | Transport protocol on which an external syslog server listens. Use 'udp' or 'tcp'. |
| Log level for remote syslog server | Log messages above this level will be sent to the external syslog server. Use one of 'emergency', 'alert', 'critical', 'error', 'warning', 'notice', 'info', 'debug'. |
| Log files rotation frequency | Sets the interval for log files rotation. Use "daily", "weekly" or "monthly". |
| Number of old log files to keep | Sets the number of rotated log files to keep before deletion. |
| Secondary remote syslog server address | Address of the secondary external syslog server. Use empty value to not use secondary external syslog server. |
| Secondary remote syslog server port | Port number on which the secondary external syslog server listens. |
| Secondary remote syslog transport | Transport protocol on which secondary external syslog server listens. Use 'udp' or 'tcp'. |
| Log level for secondary remote syslog server | Log messages above this level will be sent to secondary external syslog server. Use one of 'emergency', 'alert', 'critical', 'error', 'warning', 'notice', 'info', 'debug'. |
| Send CDRs to remote syslog server | Enables or disables including the CDR entries in the log messages sent to the remote syslog server. |

## 2.20 Signaling SSL

Table 20: Signaling SSL Parameters

| Parameter Name | Description |
|---|---|
| Revoked certificates (CRL) file | CRL file holding a list of revoked certificates. Used by sems signaling process only. |
| Minimal supported TLS version | The minimal supported TLS version on signaling interfaces. Use tls1 or tls1.1 or tls1.2. |
| TLS cipher list | The supported TLS ciphers list for signaling interfaces and proxy and similar apps on custom interfaces, openssl syntax. |
| TLS EC curves list | Allows for setting the EC curves used with TLS for signaling interface. The string is a colon separated list of curve NIDs or names, for example “P-521:P-384:P-256”. |
| Dump TLS session keys to file | If enabled, the TLS session keys will be dumped to a file for diagnostics (into directory /data/pcap/tls_keys). Disabled by default.<br>Requirements: note that this option must be enabled if one wishes to download from the GUI a bundle composed of pcap files and tls keys. Otherwise, the bundle may only contain pcap files. Limitations: WebRTC interface isn’t supported. |

## 2.21 RTP handling Parameters

Table 21: RTP handling Parameters

| Parameter Name | Description |
|---|---|
| Force symmetric RTP for mediaserver apps: | If enabled, embedded media processing actions will ignore IP addresses in callers’ SDP and send its RTP to where caller’s RTP came from. |
| RTP keep-alive frequency | Defines how often if at all ABC SBC sends RTP keep-alive packets to its peers. See rtpinactivityandkeepalive. |
| RTP timeout | Defines period of time after which a call is terminated if RTP packets stop arriving. See rtpinactivityandkeepalive. |
| Learn remote media address interval | Interval (in milliseconds) after first RTP packet received in which RTP address may still change and will be re-learned. I.e. after that interval SEMS locks on 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. Default value: 0 ms (disabled), lock on the first packet |

continues on next page

Table 21 – continued from previous page

| | |
|---|---|
| Recording playout buffer type | Type of playout buffer used for data synchronization while recording into a WAV file.<br>Possible values:<br>• adaptive<br>Sophisticated playout buffer that should be more appropriate from user's perspective, especially with higher jitter and packet loss showing in the RTP stream.<br>• simple<br>Basic buffering that might not be sufficient with lossy line.<br>Default value: adaptive |
| MOS Average S/M/L Coefficient | Coefficient for short/medium/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*. |

# 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.

Table 1: Log Level categories

| **Category** | **Subcategory** |
|---|---|
| **Core** | • Main<br>• Config<br>• Thread<br>• Timer<br>• Events<br>• SessionContainer<br>• SessionProcessor<br>• SessionWatcher<br>• MediaProcessor<br>• Plugin<br>• Utils |

continues on next page

Table 1 – continued from previous page

| | |
|---|---|
| **SIP** | • Ctrl<br>• Parser<br>• Transport<br>• Transaction<br>• Dialog<br>• OfferAnswer<br>• Session<br>• Registration<br>• Subscription<br>• DNS<br>• Blacklist |
| **B2B** | • B2BSession<br>• B2BMedia |
| **SDP** | • Parser<br>• MimeBody |
| **RTP** | • Stun<br>• RtpPacket<br>• RtcpPacket<br>• RtpTransport<br>• RtpStream<br>• RtpAudio |
| **SRTP** | • SRTP<br>• SDES<br>• DTLS<br>• ZRTP<br>• Socket |
| **AUDIO** | • Audio<br>• AudioFile<br>• AudioMixer<br>• Conference<br>• Playlist<br>• Prompt<br>• Jitter |

continues on next page

Table 1 – continued from previous page

| | |
|---|---|
| **PLUGIN** | • sbc<br>• redis_store<br>• websock<br>• reg_agent<br>• cc_gui<br>• cc_gui_rules<br>• sbc_replication<br>• webconference<br>• rest<br>• dsm<br>• xmlrpc2di |

## 3.1 Debug log level per node or per system

There is an option to set a debug level for all components either per whole system or just per single node. Debug log level will be enabled or disabled for following applications: sems, gopi, prov2json, json2redis, gogoconf, gui.

### 3.1.1 Per system

SSH to a CCM node and on a command line execute following command:

```
% sbc-toggle-debug -c -e
```

This will enable debug logging of all supported tools in whole system. To disable debug logging just execute:

```
% sbc-toggle-debug -c
```

### 3.1.2 Per node

SSH to a node for which a debug level should be set and execute following commands:

```
% sbc-toggle-debug -a -e
```

This will enable debug logging of all supported tools for that specific node. To disable debug logging just execute:

```
% sbc-toggle-debug -a
```

# 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:

- *Destination Monitor Parameters*
- *Failover Parameters*
- *Registration Agent Parameters*
- *Topology Hiding Parameters*
- *Firewall Blacklisting Parameters*
- *Security Parameters*
- *SIP Timer Parameters*
- *Resolver Parameters*

## 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 Sec-adaptive-blacklisting 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 Sec-regagent. 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.

Table 1: Registration Agent Parameters

| 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 |

continues on next page

Table 1 – continued from previous page

| | |
|---|---|
| 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.<br>Limitations:<br>• In *Lazy* mode, only the IP address is checked for existance in the cache and not port & transport.<br>• 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.<br>Available since: 5.2 |
| Bulk Contact | Turn on to support the SIP bulk contact registration form as described in RFC3680. |

## 4.4 Topology Hiding Parameters

The Section Sec-Tolplogy 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 Sec-Abuse. 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. |

## 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.<br>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.<br>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:<br>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.<br>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.<br>The same look-up logic applies to finding a destination call-agent as well.<br>This configuration is per-leg.<br>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 meetme_multi_lingual 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. |

continues on next page

Table 1 – continued from previous page

| Audio file | Content |
|---|---|
| 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. |
| 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

Table 2: 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' |

## 5.3 Two-Factor authentication

Table 3: Audio prompts

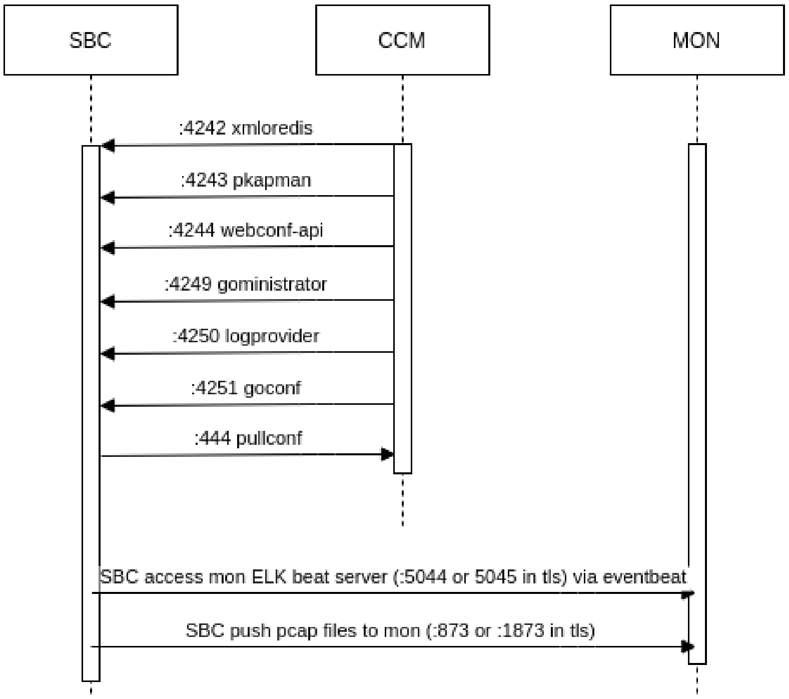
| Audio file | Context | Content |
|---|---|---|
| 2fa_greeting | Use for welcome | 'Please enter the two factor authentication PIN number that was set for this line |
| 2fa_pin_correct | Used in case of success | 'that is correct, thanks you. Please hold the line to be connected' |
| 2fa_failed | Used 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.' |

# 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 sbcinterfaceconfig), 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 Sec-uablacklisting).


SBC
CCM
MON
:4242 xmloredis
:4243 pkapman
:4244 webconf-api
:4249 goministrator
:4250 logprovider
:4251 goconf
:444 pullconf
SBC access mon ELK beat server (:5044 or 5045 in tls) via eventbeat
SBC push pcap files to mon (:873 or :1873 in tls)

| 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.<br>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. |
| 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

**Note:** 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*), reachable via *http* allowing initial configuration to be push to it. Such behavior can be tweaked in the following manner:

- disable default listening by setting the environment variable *NO_BOOTSTRAP_PULL* to *1*
- set default listening IP address by setting the environment variable *SERVER_IP* to the desired IP

For every valid configuration publish 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.<br>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. |
| Interface Options | Special interface options.<br>Note: allowed value is *force_via_address*. |
| TOS | This sets "type of service" field in IP packets header.<br>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.<br>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.<br>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:<br>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.<br>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.<br>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.<br>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 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.<br>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).<br>Note 1: disable by default.<br>Note 2: incompatible with the "default certificate" due to the CA certificate requirement.<br>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"<br>Default value: "any" |
| Capturing filter | Berkley packet filter for capture<br>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.<br>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 |
|---|---|---|
| sbc-install | initial ABC SBC installation | Sec-Install |
| sbc-backup | back up ABC SBC configuration | Sec-Recovery |
| sbc-restore | recovery of a backed up configuration | Sec-Recovery |
| sbc-set-confversion | forcibly sets config version number on config master | Sec-Recovery |
| sbc-init-config | This command configures IP address or DNS name of the main configuration node, from which ABC SBC node will automatically get configuration. It has to be run on all SBC nodes. This script is part of installation procedure. | Sec-Initial-Config-GUI |
| sbc-set-master | set up a configuration master | Sec-Initial-Config-GUI |
| sbc-publish-config | Activate the current SBC configuration and make it available for all nodes. | |
| sbc-daily-backup | Creates daily SBC backup, if enabled under Config / Global config / Backup tab. | |
| sbc-apply-config | Manually applies ABC SBC json configuration on backup node. Use –help option for command line options help. | |
| sbc-apply- provtables | Manually applies ABC SBC provisioned tables on backup. Use –help option for command line options help. | |
| sbc-passwd | Set root user password. Has to be used instead of system passwd command, to allow the password persistence when replacing container. | Sec-Initial-Config |
| 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. | |

## 9.2 User Management

| CLI | Purpose | Reference |
|---|---|---|
| sbc-add-user | Add new GUI user or add a user to a group. | Sec-User_CLI |
| sbc-del-user | Remove a GUI user or remove a user from a group. | Sec-User_CLI |
| sbc-list-groups | Get list of existing user groups | Sec-User_CLI |
| sbc-list-users | Get list of SBC users | Sec-User_CLI |
| sbc-user-passwd | Change password of SBC user, unlock user locked by too many login attempts or reset two-factor authentication secret. | Sec-User_CLI |

## 9.3 Low-Level CLI

| CLI | Purpose | Reference |
|---|---|---|
| sbc-create-config module | This command regenerates configuration files from their templates. | |
| sbc-activate-config | like *sbc-create-config all* but restart the appropriate service after the config generation | |
| 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* |
| sbc-status | Shows ABC SBC node status, which is collected automatically every minute and also shown on config master node GUI on System status page. | |
| sbc-events-queue | Show number of events waiting in redis queue on Sbc to be delivered to primary and secondary ABC Monitor. | |

## 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. | ha_statuses |

## 9.5 Other CLI

| CLI | Purpose | Reference |
|---|---|---|
| sbc-calc-ha1 | Calculates HA1.<br>Can be used to calculate parameters of *UAS auth* action. | *UAS auth* |

# 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*

## 13.1 Login

Parameters related to login/logout.

Table 1: Login Parameters

| Parameter Name | Description |
|---|---|
| GUI auto-logout time | Timeout in minutes of inactivity after which the GUI user is automatically logged out. Use '0' to disable auto-logout |
| Max failed login | Maximum number of failed logins till the user account is blocked. This is for brute force hacking protection. Use '0' to disable account blocking due to failed logins. |
| Blocking period | How long the user account is blocked (in seconds) if number of invalid logins reach the 'Max failed login' |
| Allow concurrent login | Concurrent login of single GUI user from multiple devices is not allowed by default. Checking this checkbox will allow it. |
| Garbage collect timeout | Timeout (in days) after which the data used for brute force hacking protection are removed from DB. |
| Do not allow re-use passwords - history length | If this option is set, users are not allowed to set a new password that is the same as any of the last passwords he or she has used. This field set number of passwords that are checked. |
| Password expiration (days) | Number of days in which user password expire and have to be changed. Set to zero to never expire. |
| Minimum password length | The minimum length of user password. |
| Password strength policy | Define set of characters that have to be present in user password. |
| Enable two factor authentication for LDAP users (initial value) | When checked, LDAP users newly connected to the CCM will have two-factor authentication enabled. |
| Relying Party ID for passkeys | This is Relying Party Identifier identifying the WebAuthn Relying Party on whose behalf a given registration or authentication ceremony is being performed. A public key credential can only be used for authentication with the same entity (as identified by RP ID) it was registered with. By default, the RP ID for a WebAuthn operation is set to the caller's origin's effective domain. See: https://www.w3.org/TR/webauthn/#relying-party-identifier for more details. |

## 13.2 LDAP Parameters

Cluster Config Manager GUI allow a two step authentication against an LDAP server. The first authentication, "LDAP auth", check a user against the LDAP server. Here, the user dn (*uid=john,ou=People,dc=example,dc=org*) and it's password (*johnldap*) are used. The second check, "GUI auth", ensure that at least one of the LDAP user groups' match one of the GUI capability ABC SBC groups.

Once configured, user wishing to login can use their LDAP UIDs and password onto the Cluster Config Manager log page.

Table 2: LDAP Parameters

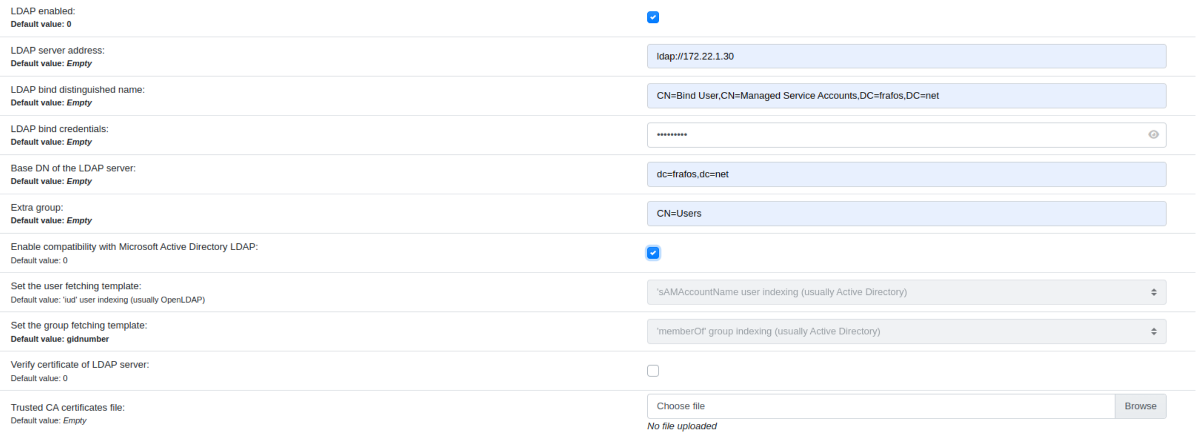
| Parameter Name | Description |
|---|---|
| LDAP auth enabled | Enable LDAP authentication. |
| LDAP server address | LDAP host on which the LDAP service can be reached (ldap://IP:PORT or ldap://IP or ldap://my.domain) |

continues on next page

Table 2 – continued from previous page

| | |
|---|---|
| LDAP distinguished name / admin user DN | Specifies the distinguished name used to bind to the LDAP server for lookups. |
| LDAP credentials / admin user PW | Specifies the LDAP credentials used to bind. |
| base DN such as 'dc=example,dc=org' | Default search DN of the LDAP.<br>Ex: For "cn=admin,dc=example,dc=org", base DN is "dc=example,dc=org" |
| extra group such as 'ou=People' like in "uid=john,ou=People ,dc=example,dc=org" | So user only need to register their name (aka "uid") please pass any extra bind dn via this parameters.<br>Ex: user (like *john*) exist in the form, "uid=john,ou=People,dc=example,dc=org", so we set the following to "ou=People". GUI will then concatenate in the form uid=[user value][extra_group][base_dn] to auth the user against the ldap server.<br>Note that to complete a user login, the ldap user must **also** be member of a group matching one of the GUI groups supporting login. This group must be a primary group of that user. |
| Enable Active compatibility with Microsoft Active Directory LDAP | Connect to an Active Directory LDAP server. |
| User template | Please select according to your LDAP configuration. Microsoft Active Directory users should select 'sAMAcountName'. Usual OpenLDAP configuration use 'uid', but some setup rely on 'cn'. |
| Group template | Please select according to your LDAP configuration. Microsoft Active Directory users should select 'memberOf'. Usual OpenLDAP configuration use 'gidNumber', but some setup rely on 'memberUid'. |
| Verify certificate of LDAP server | |
| Trusted CA certificates file | Select a file containing list of certificates to which the client's one are check. The certificate must be in PEM format. Use an Active Directory LDAP server. |

Example of an OpenLDAP configuration:



| | |
|---|---|
| LDAP enabled:<br>**Default value: 0** | ☑ |
| LDAP server address:<br>**Default value: *Empty*** | ldap://172.22.1.30 |
| LDAP bind distinguished name:<br>**Default value: *Empty*** | CN=Bind User,CN=Managed Service Accounts,DC=frafos,DC=net |
| LDAP bind credentials:<br>**Default value: *Empty*** | •••••••••• |
| Base DN of the LDAP server:<br>**Default value: *Empty*** | dc=frafos,dc=net |
| Extra group:<br>**Default value: *Empty*** | CN=Users |
| Enable compatibility with Microsoft Active Directory LDAP:<br>Default value: 0 | ☑ |
| Set the user fetching template:<br>Default value: 'iud' user indexing (usually OpenLDAP) | 'sAMAccountName user indexing (usually Active Directory) |
| Set the group fetching template:<br>**Default value: gidnumber** | 'memberOf' group indexing (usually Active Directory) |
| Verify certificate of LDAP server:<br>Default value: 0 | □ |
| Trusted CA certificates file:<br>Default value: *Empty* | Choose file Browse<br>*No file uploaded* |

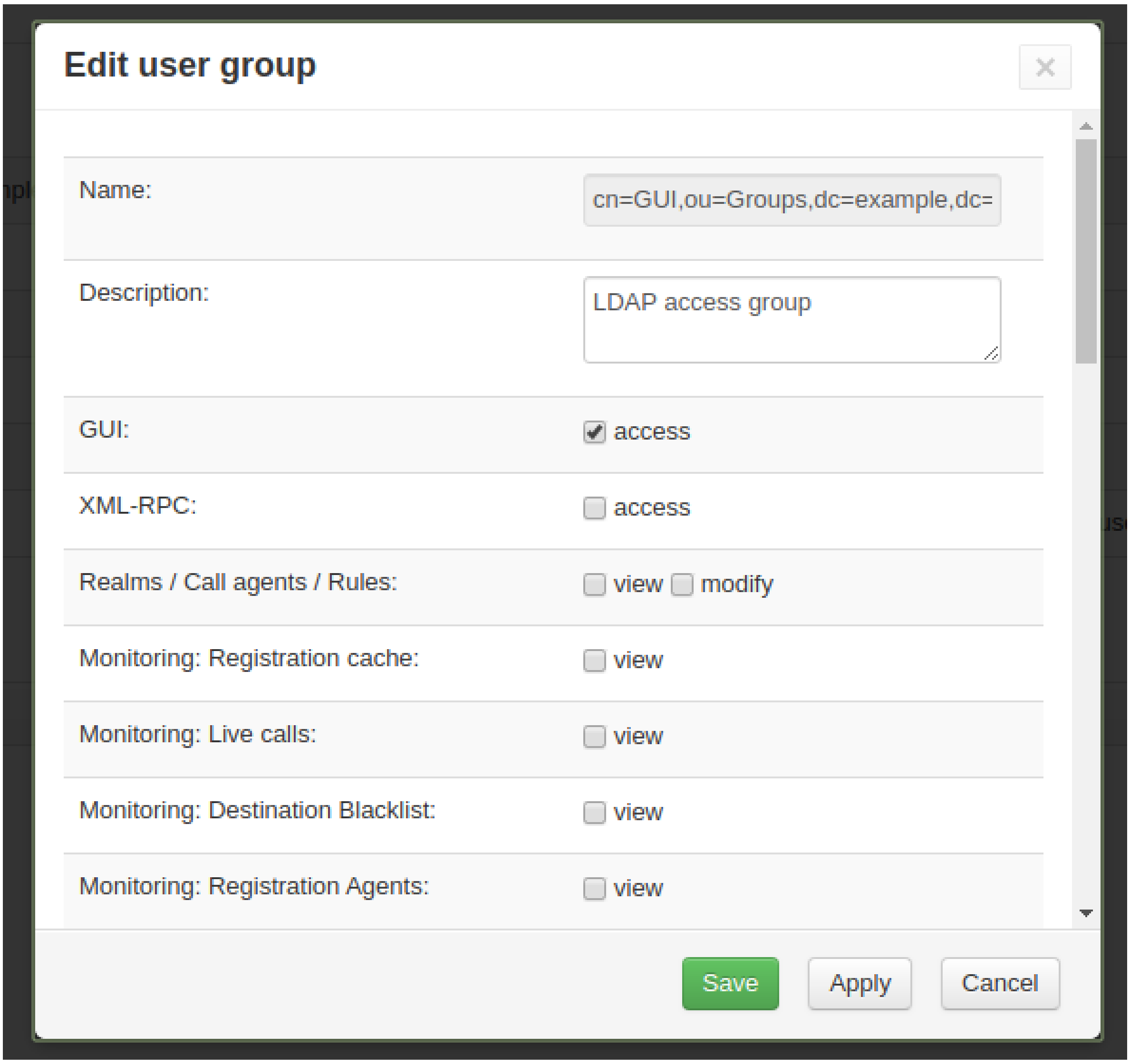
There is a docker container available on github that match the screenshot configuration : https://github.com/frafos/docker-ldap.

The image come in with 2 users (+ admin) :

| User | dn | pwd | note |
|---|---|---|---|
| john | *uid=john,ou=People, dc=example,dc=org* | *johnldap* | The following example work for that user. |
| jane | *uid=jane,ou=People, dc=example,dc=org* | *janelday* | The following example **doesn't** work for that user. John and Jane belongs to different groups. |

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*) :



Edit user group

Name: cn=GUI,ou=Groups,dc=example,dc=

Description: LDAP access group

GUI: ☑ access

XML-RPC: □ access

Realms / Call agents / Rules: □ view □ modify

Monitoring: Registration cache: □ view

Monitoring: Live calls: □ view

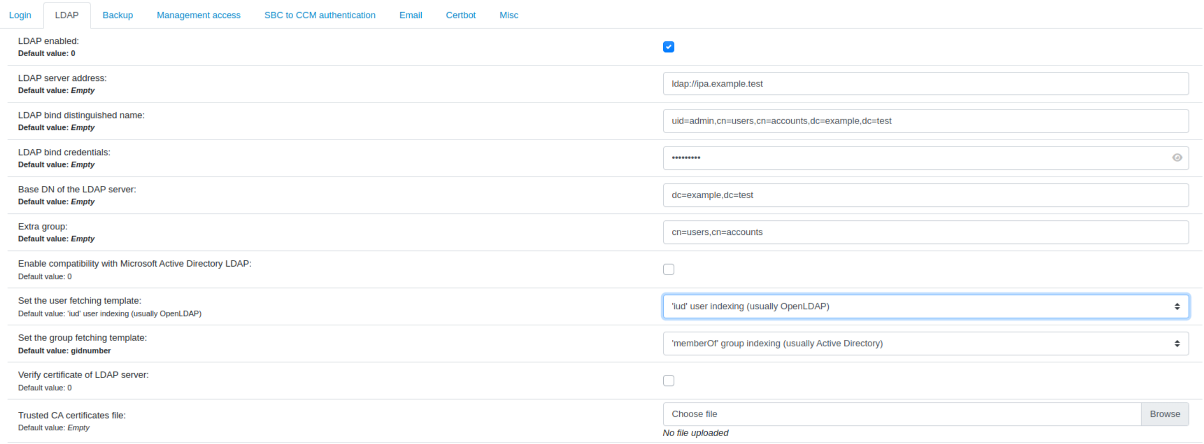
Monitoring: Destination Blacklist: □ view

Monitoring: Registration Agents: □ view

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

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).

Example of a FreeIPA LDAP configuration:



| | |
|---|---|
| LDAP enabled:<br>**Default value: 0** | ☑ |
| LDAP server address:<br>**Default value: *Empty*** | ldap://ipa.example.test |
| LDAP bind distinguished name:<br>**Default value: *Empty*** | uid=admin,cn=users,cn=accounts,dc=example,dc=test |
| LDAP bind credentials:<br>**Default value: *Empty*** | •••••••• |
| Base DN of the LDAP server:<br>**Default value: *Empty*** | dc=example,dc=test |
| Extra group:<br>**Default value: *Empty*** | cn=users,cn=accounts |
| Enable compatibility with Microsoft Active Directory LDAP:<br>Default value: 0 | □ |
| Set the user fetching template:<br>Default value: 'iud' user indexing (usually OpenLDAP) | 'iud' user indexing (usually OpenLDAP) |
| Set the group fetching template:<br>**Default value: gidnumber** | 'memberOf' group indexing (usually Active Directory) |
| Verify certificate of LDAP server:<br>Default value: 0 | □ |
| Trusted CA certificates file:<br>Default value: *Empty* | Choose file<br>*No file uploaded* |

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

In our case, the Free IPA server was configured with defaults 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
```

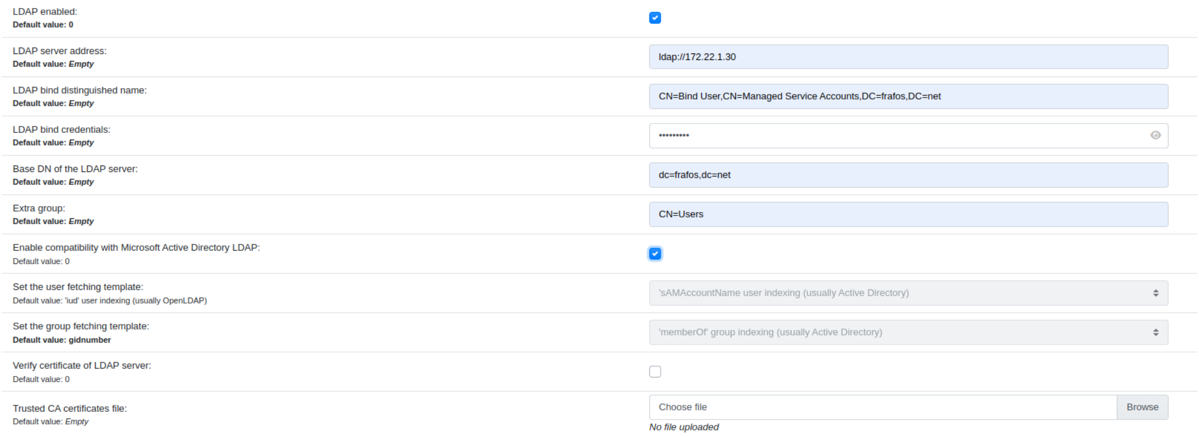
(continues on next page)

(continued from previous page)

```
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.

Example of an Microsoft Active Directory configuration:



| | |
|---|---|
| LDAP enabled:<br>**Default value: 0** | ☑ |
| LDAP server address:<br>**Default value: *Empty*** | ldap://172.22.1.30 |
| LDAP bind distinguished name:<br>**Default value: *Empty*** | CN=Bind User,CN=Managed Service Accounts,DC=frafos,DC=net |
| LDAP bind credentials:<br>**Default value: *Empty*** | ••••••••• |
| Base DN of the LDAP server:<br>**Default value: *Empty*** | dc=frafos,dc=net |
| Extra group:<br>**Default value: *Empty*** | CN=Users |
| Enable compatibility with Microsoft Active Directory LDAP:<br>Default value: 0 | ☑ |
| Set the user fetching template:<br>Default value: 'iud' user indexing (usually OpenLDAP) | 'sAMAccountName user indexing (usually Active Directory) |
| Set the group fetching template:<br>**Default value: gidnumber** | 'memberOf' group indexing (usually Active Directory) |
| Verify certificate of LDAP server:<br>Default value: 0 | ☐ |
| Trusted CA certificates file:<br>Default value: *Empty* | Choose file Browse<br>*No file uploaded* |

## 13.3 Backup Parameters

These parameters set ABC SBC daily backups. See also more in Sec-Backup.

Table 3: Backup Parameters

| Parameter Name | Description |
|---|---|
| Create daily Sbc configuration backups | If enabled, daily snapshot of ABC SBC configuration will be created into backup gzipped tarball file. |
| Include provisioned tables in daily or automatic backups | If enabled, the daily or automatic backup will include also content of whole provisioned tables. The automatic backup is created when new container is started and database is going to be upgraded, for possible restore in case of switch back to older container. |
| Number of days to keep backups | Sets the retention period for backup files. All files named sbc-backup-* in the backup directory older than specified number of days will be deleted on every daily backup run. Use 0 to disable automatic deletion of old backup files. |
| Destination directory for backups | Specifies the destination directory for the daily backup files. Default is "/data/backups" directory. |
| Full path to extra files or dirs to include in backup | Extra custom files or dirs to include in backup can be listed using full path, more fields separated by comma. A * wildcard can be used. The path must not contain comma character. |

## 13.4 Management access Parameters

Table 4: Management access Parameters

| Parameter Name | Description |
|---|---|
| SSL certificate file for GUI, REST API and XML-RPC interface | Select a file containing SSL certificate in PEM format. |
| SSL private key file for GUI, REST API and XML-RPC interface | Select a file containing key for SSL certificate in PEM format. |
| TLS cipher list | The supported TLS cipher list for GUI, xmlrpc and config pull, in openssl syntax. |
| Minimal supported TLS version for GUI, REST API and XML-RPC interface | Select the minimal TLS version that should be supported on GUI, REST API and XML-RPC interfaces. |

## 13.5 SBC security Parameters

These parameters are used to authenticate SBCs to CCM on services running on IMI interface like Pullconf, Local monitoring query service, Management for host and other services.

Table 5: SBC security Parameters

| Parameter Name | Description |
|---|---|
| HTTP Basic Authentication username for configuration pull or status push | Username that SBC nodes are using to pull the configuration from the CCM. The same one will need to be set in the SBC node in sbc-init-config. |
| HTTP Basic Authentication password for configuration pull or status push | Password that SBC nodes are using to pull the configuration from the CCM. The same one will need to be set in the SBC nodes in sbc-init-config. |
| SSL certificate file for pull-conf | Select a file containing SSL certificate in PEM format (Without password). |
| SSL private key file for pullconf | Select a file containing key for SSL certificate in PEM format (without password). |
| Trusted CA certificates file | Select a file containing list of certificates against which the clients' certs are checked. If intermediate CAs are used, the whole chain needs to be in this file. The certificates must be in PEM format (without password). |
| Enable mTLS | If checked, the CCM verifies the TLS certificate of the peer against the trusted CA certificates. |

## 13.6 Email Parameters

These parameters are used to configure sending emails from CCM.

Table 6: CCM Email Parameters

| Parameter Name | Description |
|---|---|
| Email address for sending certificate and other alerts | Email address to which important alerts like certificate renewal failure, acquisition success and other are sent. Field is required to be set if any Let's Encrypt certificate is expected to be used |
| From email address for sending alerts | Email address used for From in email alerts, system default is used if empty. |
| SMTP email server address server for sending alerts | Set the SMTP server address, that emails from CCM will be sent to. Note: when ABC SBC is running in container, mail relay on localhost is not available and external mail server has to be used. |
| SMTP mail server port | Set the SMTP mail server port. |
| Use secure connection to SMTP mailserver | Set if the SMTP connection to mailserver should be encrypted using TLS or STARTTLS. |
| **SMTP mail server**<br>authentication | Use 'off' to disable the authentication, or 'on' to enable it and choose auth type automatically. |
| Username for SMTP authentication. | Set the username for SMTP authentication, if authentication is enabled. |
| Password for SMTP authentication | Set the password for SMTP authentication, if authentication is enabled. |

## 13.7 Certbot Parameters

Cluster Config Manager' certbot act like the famous Let's Encrypt certbot. For more information, please referee to the TLS' chapter letsencrypt.

Table 7: Certbot Parameters

| Parameter Name | Description |
|---|---|
| Query Let's Encrypt staging environment | In case of testing, we recommend querying the staging environment to avoid reaching Let's Encrypt 168h rate limit.<br>Please note that staging certificates are not suitable for productions. |
| Attempt renewal X days before certificate expiration | By default, the certbot attempts to renew a certificate 15 days before it expires.<br>Please note that this setting doesn't affect automatic email notifications about certificate expiration from Let's Encrypt. |
| CRON job interval | Set CRON job interval rule (in CRON format), allowing refinement for the interval at which the certbot is automatically run in an attempt renew near expiration certificates. |

Please note that the certbot is invoked under the following condition: - by CRON job call, every night a 1am - when a node successfully pull a new configuration - when a configuration has successfully been pushed to a node

You may manually invoke the certbot, from within a Cluster Config Manager' shell by running the following:

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

In case of testing, to avoid reaching LE' 168h rate limit, please remember to enable the "Query Let's encrypt staging environment" Cluster Config Manager' config options.

## 13.8 Miscellaneous Parameters

Table 8: Miscellaneous Parameters

| Parameter Name | Description |
|---|---|
| Automatically add new nodes | If enabled, records for new nodes that pull config from configuration master will be automatically added. If disabled, the configuration master will refuse to provide configuration to nodes that are not already defined in Nodes configuration. |
| Compatibility mode | When using CCM with older SBCs, it is possible to select the SBC version here. The CCM will then hide settings (e.g. rule conditions and actions, interface applications, global config values or entire screens) that are unavailable in the selected SBC version.<br>Selecting the 'No limitation' option will displays all settings for all supported SBC versions.<br>The default value for this option is the latest SBC version. As some interface applications and other features were removed in SBC 5.5, these settings are no longer shown by default. |
| Compatibility mode with secunet SBC | If enabled, the firewall control and HA configuration screens will be hidden. |
| Allow overlap of Call Agent IP ranges | If enabled, GUI will not check whether ranges of IP addresses of call agents are same or overlapped. |
| 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. |
| 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. |

# Chapter 14

# 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 15

# 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 16

# Legacy application

## 16.1 SSH

**Important:** 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. |

## 16.2 SNMP

**Important:** 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. |

## 16.3 TURN server for websocket

**Note:** Application available since SBC release 4.5.

**Important:** 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. |

## 16.4 Local monitoring query service

**Note:** Application available since SBC release 4.2.

**Important:** 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.<br>Note: value not editable (*4242*). |

## 16.5 PCAP query service

**Note:** Application available since SBC release 4.5.

**Important:** 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.<br>Note: value not editable (*4243*). |

## 16.6 Local webconf API

**Note:** Application available since SBC release 4.6.

**Important:** 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.<br>Note: value not editable (*4244*). |

## 16.7 Management for host

**Note:** Application available since SBC release 4.5.

**Important:** Replaced by *Unified SBC management service* starting 5.5.

The *sbc-gominstrator* 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.<br>Note: value not editable (*4249*). |

## 16.8 Log files provider

**Note:** Application available since SBC release 5.1.

**Important:** 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.<br>Note: value not editable (*4250*). |

Application is available since ABC SBC' release 5.1.

## 16.9 Local packet classifier

**Note:** Application available since SBC release 5.4.

**Important:** 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.<br>Note: value not editable (*4252*). |

## 16.10 HTTP proxy

**Note:** Application available since SBC release 4.6.

**Important:** 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.<br>Note: mandatory. |
| Target port | Port to which the proxy redirect.<br>Note: mandatory. |
| TLS enable | Proxy over TLS. |

## 16.11 HTTP redirect

**Note:** Application available since SBC release 4.6.

**Important:** 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.<br>Note: mandatory. |
| TLS enable | Redirect over TLS. |

## 16.12 Prometheus Pull Service

**Note:** Application available since SBC release 5.2.

**Important:** 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. |

# Index