SBC Ruleset Configuration

DGW 50.2

2026-03-31

Table of Contents

SBC Conditions

This section describes the conditions that can be used in SBC's Call Agent or Routing Rules.

Name	Description	Operators
Source Call Agent	Checks the source Call Agent.	**==** , **!=**
Source IP	Checks the IP address the incoming request was sent from.	**==** , **!=** , **RegExp** , **does not match RegExp** , **begins with** , **does not begin with**
Source Port	Checks the port number the incoming request was sent from.	**==** , **!=** , **RegExp** , **does not match RegExp** , **begins with** , **does not begin with**
Inbound interface	Checks the local interface the incoming request was received from. Value is a Signaling Interface name.	**==** , **!=** , **RegExp** , **does not match RegExp** , **begins with** , **does not begin with**
R-URI	Checks the request URI.	**==** , **!=** , **RegExp** , **does not match RegExp** , **begins with** , **does not begin with**
R-URI user	Checks the user part of the request URI.	**==** , **!=** , **RegExp** , **does not match RegExp** , **begins with** , **does not begin with**

Name	Description	Operators
R-URI user parameter	Checks the parameter(s) in the username part of the request URI For example in the R-URI "sip:106;name=smith@domain.com", the parameter "name" can be checked for value "smith".	**==** , **!=** , **RegExp** , **does not match RegExp** , **begins with** , **does not begin with**
R-URI domain	Checks the host part of the request URI. The host part can contain a port number.	**==** , **!=** , **RegExp** , **does not match RegExp** , **begins with** , **does not begin with**
R-URI URI parameter	Checks the parameter(s) of the request URI.	**==** , **!=** , **RegExp** , **does not match RegExp** , **begins with** , **does not begin with**
From	Checks the 'From' header value.	**==** , **!=** , **RegExp** , **does not match RegExp** , **begins with** , **does not begin with**
From URI	Checks the value of the 'From' header URI.	**==** , **!=** , **RegExp** , **does not match RegExp** , **begins with** , **does not begin with**

Name	Description	Operators
From user	Checks the user part of the 'From' header URI.	**==** , **!=** , **RegExp** , **does not match RegExp** , **begins with** , **does not begin with**
From domain	Checks the host part of the 'From' header URI. The host part can contain a port number.	**==** , **!=** , **RegExp** , **does not match RegExp** , **begins with** , **does not begin with**
To	Checks the value of the 'To' header.	**==** , **!=** , **RegExp** , **does not match RegExp** , **begins with** , **does not begin with**
To URI	Checks the value of the 'To' header URI.	**==** , **!=** , **RegExp** , **does not match RegExp** , **begins with** , **does not begin with**
To user	Checks the user part of the 'To' header URI.	**==** , **!=** , **RegExp** , **does not match RegExp** , **begins with** , **does not begin with**

Name	Description	Operators
To domain	Checks the host part of the 'To' header URI. The host part can contain a port number.	**==** , **!=** , **RegExp** , **does not match RegExp** , **begins with** , **does not begin with**
Header	Checks the value of a given SIP header.	**==** , **!=** , **RegExp** , **does not match RegExp** , **begins with** , **does not begin with**
Codecs	Checks the presence/absence of specified codecs within de SDP payload. Codecs name must be specified according to IETF RTP payload types.	**Contain** , **Do not contain** , **Contain RegExp**
Media Types	Checks the presence/absence of a media type within the SDP. Right operand specifies the media type name (audio, video, image or text).	**Contain** , **Do not contain** , **Contain RegExp**
Call parameter	Checks the value of a call parameter using the selected operator. The call parameter must first be defined by the 'Set Call parameter' action. Any condition referring to an undefined value returns FALSE as result.	**==** , **!=** , **RegExp** , **does not match RegExp** , **begins with** , **does not begin with**
Call parameter existence	Tests if a call parameter exists or not. This is useful to accurately discriminate non-existing and empty values.	**Exists** , **Does not exist**
Method	Checks the SIP request method. The value is chosen from a list of allowed methods.	**==** , **!=**

Name	Description	Operators
Register cache	Checks the content of the registration cache.	**From URI (AoR + Contact + IP/port)** , **From URI (AoR + IP/port)** , **Contact URI (Contact + IP/port)** , **To URI (AoR)** , **R-URI (Alias)**
NAT	Checks if whether or not the sender is behind the NAT. Works only if the UA directly communicates with the SBC service.	**First Via Address**
Last Action Result	Returns true if the last action is completed successfully, false otherwise.	**==**
Blacklist	Checks if a Call Agent is on a blacklist (or not). A Call Agent is blacklisted when it is not reachable to make sure that no attempts to send traffic are done. If the value of the **CallAgent.PeerHost** parameter is an IP address without a port, this condition checks if the Call Agent is blacklisted on either standard SIP port (5060 and 5061). If either port is blacklisted, the Call Agent is considered blacklisted. See also **Sbc.PeerMonitoring**.	**Call Agent Blacklisted** , **Call Agent not Blacklisted**
Parallel Call Count	Tests if the number of parallel calls is below or above a threshold. The value refers to the number on ongoing calls that were evaluated by this rule, for a specific Call Agent. It does not refer to a global number of calls.	**Is below** , **Is equal or above**

SBC Operators

This section describes the operators that can be used with conditions in the SBC Call Agent or Routing Rules.

Name	Description
!=	The left operand does not equal the specified value.
==	The left operand equals the specified value.
Call Agent Blacklisted	The Call Agent specified with the right operand is blacklisted. If the value of the **CallAgent.PeerHost** parameter is an IP address without a port, this condition checks if the Call Agent is blacklisted on either standard SIP port (5060 and 5061). If either port is blacklisted, the Call Agent is considered blacklisted. See also **Sbc.PeerMonitoring**.
Call Agent not Blacklisted	The Call Agent specified with the right operand is not blacklisted.
Contact URI (Contact + IP/ port)	A user with specified 'Contact' URI is registered from specified IP:port.
Contain	The right operand is contained in.
Contain RegExp	Sample described by the right operand is contained in.
Do not contain	The right operand is not contained in.
Does not exist	The left operand does not exist within the specified string.
Exists	The left operand exists within the specified string.
First Via Address	Compares the SIP message source IP with the the first Via address.
From URI (AoR + Contact + IP/ port)	The user with matching 'From' URI and 'Contact' URI is registered from specified IP:port.
From URI (AoR + IP/port)	The user with matching 'From' URI is registered with any 'Contact' URI from specified IP:port.
Is below	The left operand is lower than the right operand.
Is equal or above	The left operand is higher or equal than the right operand.
R-URI (Alias)	The user with specified request URI is registered.
RegExp	The left operand matches the specified regular expression. The regular expression syntax is standard except that it supports the use of replacements. When a regular expression contains a '$' character that is not used to identify a replacement, then the '$' and backslash characters must be escaped with a backslash.
To URI (AoR)	The user with specified 'To' URI is registered.
begins with	The left operand starts with the specified string.
does not begin with	The left operand does not start with the specified string.

Name	Description
does not match RegExp	The left operand does not match the specified regular expression. The regular expression syntax is standard except that it supports the use of replacements. When a regular expression contains a '$' character that is not used to identify a replacement, then the '$' and backslash characters must be escaped with a backslash.

SBC Call Agent Rule Actions

SIP Mediation Actions

Manipulation of Identities, URIs, header fields and response codes.

Set RURI

Sets the request URI to a new value

Parameter	Description
new URI	The value of the new SIP request URI.

Prefix RURI user

Adds a prefix to the user part of the request URI.

Parameter	Description
prefix string	A value to insert as a prefix to the user part of the request URI.

Set RURI user

Replaces the user part of the request URI.

Parameter	Description
new userpart	Value of the new userpart to be used in the request URI field.

Append to RURI user

Adds a suffix to the user part of the request URI. Several suffixes can be added if the action is used several times.

Parameter	Description
suffix	Suffix to add to the request URI.

Strip RURI user

Removes the leading characters of the user part of the request URI.

Parameter	Description
number of leading characters	The number of leading characters to remove.

Set RURI host

Replaces the host part of the request URI.

Parameter	Description
new hostpart	The value of the new host part.

Set RURI parameter

Sets the request URI parameter.

Parameter	Description
parameter name	Parameter name
parameter value	Parameter value

Set From

Replaces the value of the 'From' header field.

Parameter	Description
From HF value	New value for the 'From' header field.

Set From display name

Replaces the display name part of the 'From' header field.

Parameter	Description
new From display name	New value for the display name.

Set From user

Replaces the user part of the 'From' URI.

Parameter	Description
new From userpart	New user part for the 'From' URI.

Set From host

Replaces the hostname of the 'From' URI.

Parameter	Description
new From hostname	New host name for the 'From' URI.

Set To

Replaces the 'To' header field value.

Parameter	Description
To HF value	The new value of the 'To' header field.

Set To display name

Replaces the display name part of the 'To' header field.

Parameter	Description
new To display name	The new value for the 'To' display name.

Set To user

Replaces the user part of 'To' URI.

Parameter	Description
new To userpart	The new user part of the 'To' URI.

Set To host

Replaces the host name of the 'To' URI.

Parameter	Description
new To hostname	The new host name of the 'To' URI.

Remove Header

Removes all occurrences of a specified Header from all requests and responses in a session. Headers added in subsequent rules are not removed by this action.

Note: Mandatory header fields (CallID, From, To, CSeq, Via, Route, Record-Route and Contact) are not removed by this action.

Parameter	Description
header field name	Name of the header to remove. This parameter is case-insensitive. Headers in compact name form must be removed separately.

Add Header

Adds a new header field to a request.

Parameter	Description
HF name	Header field name to add.
HF value	Value of the new header field.

Add Dialog contact parameter

Allows parameters to be added to the 'Contact' URI generated by the SBC on either side of the call.

Parameter	Description
Leg	Select inbound or outbound leg.
User part	If this option is enabled, the parameter will be added to the user part of the Contact-URI.
parameter name	Parameter name to add to the 'Contact' URI.
parameter value	Value of the new parameter.

Set Header whitelist

Removes all headers that are not specified in a list of headers. The headers are removed from all requests and responses in a session.
Headers are removed after all inbound and outbound rules have been processed.

Note: Mandatory header fields (CallID, From, To, CSeq, Via, Route, Record-Route and Contact) are not removed by this action.

Parameter	Description
comma-separated header-field name list	List of header field names. Names are comma-separated and case-insensitive. Names in compact form must be listed explicitly.

Set Header blacklist

Removes all occurrences of a list of headers from all requests and responses in a session.
Headers are removed after all inbound and outbound rules have been processed.

Note: Mandatory header fields (CallID, From, To, CSeq, Via, Route, Record-Route and Contact) are not removed by this action.

Parameter	Description
comma-separated header-field name list	List of header field names. Names are comma-separated and case-insensitive. Names in compact form must be listed explicitly.

Enable SIP Session timer callee-leg

Enforces the use of a session timer on the outbound leg.

Parameter	Description
session expiration(sec)	SIP session expiration value in seconds.
minimum expiration(sec)	The minimum value for the session interval, in units of delta-seconds. When used in an INVITE or UPDATE request, it indicates the smallest value of the session interval that can be used for that session. When present in a request or in a response, the session interval MUST NOT be less than 90 seconds.

Enable SIP Session timer caller-leg

Enforces the use of a session timer on the inbound leg.

Parameter	Description
session expiration(sec)	SIP session expiration value in seconds.
minimum expiration(sec)	The minimum value for the session interval, in units of delta-seconds. When used in an INVITE or UPDATE request, it indicates the smallest value of the session interval that can be used for that session. When present in a request or in a response, the session interval MUST NOT be less than 90 seconds.

Translate reply code

Translates a SIP reply code to another value and replaces the reason text.

Parameter	Description
matching reply code	SIP reply code to be replaced.
new reply code and reason phrase	New reply code and reason phrase.

Enable transparent dialog IDs

Enforces the use of the same dialog IDs on both legs of a call.

Parameter	Description
To-tag	Method of To-tag propagation.

Forward Via-HFs

Forces the SBC to keep the Via header fields when forwarding the request.

UAC auth

Responds to authentication requests using the specified credentials.

Parameter	Description
username	Authentication user name.
password	Authentication password.
towards	Towards caller or callee.

Diversion to history-info

Converts the SIP diversion header-field into the History-Info header-field by RFC 6044.

Call Transfer Handling

Defines the mode in which REFERs are handled: rejection, local processing, or forwarding.
Note: When this ruleset action is not used, the default REFER Processing Mode is 'REFER pass-through'.

Parameter	Description
mode	REFER Processing Mode. The choices available are 'REFER pass-through', 'Handle REFER internally' and 'Reject REFER'.
bye delay	Delay, in seconds, before issuing BYE to disconnect the original call leg. Applies only when the local processing mode is used.
remove allow refer	Remove the REFER method from the Allow header in requests and responses sent to the call agent's peer.

Refer-To URI translation

Translates the Refer-To URI of a REFER if it corresponds to an SBC generated Contact back into the original Contact.
This action must be executed at several places for the feature to work properly:

- All calls and requests that might send a REFER for which the translation is needed.
- Calls to which the Refer-To points to.

Set Contact Header parameter whitelist

Sets the whitelist of the Contact Header parameters.

Parameter	Description
comma-separated parameter name list	Comma-separated list of parameters to whitelist. Only the parameters defined in this list are relayed.

Set Contact Header parameter blacklist

Sets the blacklist of the Contact Header parameters.

Parameter	Description
comma-separated parameter name list	Comma-separated list of parameters to blacklist. Parameters defined in this list are not relayed.

Forward Contact Header parameters

Forwards the Contact Header parameters.

Set Content-Type whitelist

Keeps all message bodies whose content is in the whitelist Content-Type list; all other message bodies are removed. Multipart bodies are only removed when each single part inside has been removed. This field is case-insensitive.

Parameter	Description
comma-separated Content-Type names list	Specify the Content-Type names as comma-separated list, e.g. 'application/sdp,application/my-proprietary-ct'.

Set Content-Type blacklist

Removes all message bodies whose content is in the blacklist Content-Type list; all other message bodies are kept. Multipart bodies are only removed when each single part inside has been removed. This field is case-insensitive.

Parameter	Description
comma-separated Content-Type names list	Specify the Content-Type names as comma-separated list, e.g. 'application/sdp,application/my-proprietary-ct'.

Handle INVITE with Replaces header

Handle Replaces header in an incoming INVITE: incoming call is connected to an existing call leg.

SDP Mediation Actions

Manipulation of codec and early media negotiation.

Set CODEC whitelist

Removes all but listed codecs from the SDP payload.
If no media type is left in the SDP payload after applying the action, i.e. removing the non-listed codecs, the request is rejected with a 488 SIP response.

Parameter	Description
Value	Comma-separated list of codecs to be accepted in the SDP payload. Codec names are case-insensitive and must be specified according to IETF RTP payload types.

Set CODEC blacklist

Removes all listed codecs from the SDP payload.
If no media type is left in the SDP payload after applying the action, i.e. removing the listed codecs, the request is rejected with a 488 SIP response.

Parameter	Description
Value	Comma-separated list of codecs to remove from the SDP payload. Codec names are case-insensitive and must be specified according to IETF RTP payload types.

Set CODEC preferences

Promotes specified codecs in SDP negotiation.

Parameter	Description
inbound leg	Comma-separated list of codecs to promote in the SDP payload of the inbound leg. Codec names are case-insensitive and must be specified according to IETF RTP payload types.
outbound leg	Comma-separated list of codecs to promote in the SDP payload of the outbound leg. Codec names are case-insensitive and must be specified according to IETF RTP payload types.

Set Media whitelist

Removes all but listed media types from the SDP payload.
If no media type is left in the SDP payload after applying the action, i.e. removing the non-listed media types, the request is rejected with a 488 SIP response.

Parameter	Description
Value	Comma-separated list of media types (audio, video, image, text...) to be accepted in the SDP.

Set Media blacklist

Removes all listed media types from the SDP payload.
If no media type is left in the SDP payload after applying the action, i.e. removing the listed media types, the request is rejected with a 488 SIP response.

Parameter	Description
Value	Comma-separated list of media types (audio, video, image, text...) to be remove from the SDP.

Set SDP attributes whitelist

Sets the whitelist of SDP attributes.

Parameter	Description
comma-separated sdp attributes list	Comma-separated list of SDP attributes to whitelist. Only the parameters defined in this list are relayed. This whitelist has no effect on: • media direction attributes (sendrecv/sendonly/recvonly/inactive) • payload specific attributes (rtpmap/fmtp).

Set SDP attributes blacklist

Sets the blacklist of SDP attributes.

Parameter	Description
comma-separated sdp attributes list	Comma-separated list of SDP attributes to blacklist. Parameters defined in this list are not relayed. This blacklist has no effect on: • media direction attributes (sendrecv/sendonly/recvonly/inactive) • payload specific attributes (rtpmap/fmtp).

Drop early media

Drop RTP frames until the call is established.

Drop SDP from 1xx replies

Removes SDP payload from specific 1xx responses.

Parameter	Description
affected replies	Comma-separated list of SIP response codes.

Rectify SDP content

Rectify the content of relayed SDP payload.

Parameter	Description
Handle missing stream direction	Add explicitely the stream direction in relayed SDP payload. When disabled, the absence of stream direction may be considered as "a=sendrecv" by SIP peers. This is as RFC3264. When enabled, if the SDP answer to relay is positive, the absence of stream direction is detected and the relayed SDP will contain the matching stream direction: • The sent SDP offer contained "a=inactive", the relayed SDP answer will contain "a=inactive". • The sent SDP offer contained "a=sendrecv", the relayed SDP answer will contain "a=sendrecv". • The sent SDP offer contained "a=recvonly", the relayed SDP answer will contain "a=sendonly". • The sent SDP offer contained "a=sendonly", the relayed SDP answer will contain "a=recvonly".
Hold IP Connection Address	Rectify the Connection Address when its IP address is 0.0.0.0. The possibilities are: • No Action: The value of the Connection Address is not rectified. • Replaced by the RTP Anchoring IP: The advertised IP used to relay the media is used. This option requires having the "Enable RTP anchoring" action.

Management and Monitoring Actions

Logging packets and custom notifications.

Log received traffic

Logs SIP/RTP traffic into a capture file.
The capture file (in PCAP format) is stored in the 'sbc/logs' folder of the File service.

Note: The Log received traffic action can only be used in inbound rules.

Parameter	Description
Log type	Selects the type of packets to include in the capture file. • SIP only: Include SIP messages only. • SIP and RTP: Include SIP messages and RTP frames.
Show DNS queries	Show DNS queries in PCAP file
PCAP file name	Represents the name of the PCAP file. This name MUST be unique for each call. In order to link the PCAP file to a specific call, it is recommended to use the $ci-$ft expression as the PCAP file name. Using the expression creates a file name containing the SIP Call-ID and From-tag used to identify a unique call. When the PCAP file name parameter is empty, a unique file name is automatically generated. However, in this case, it is not possible to link the PCAP file to a specific call as the PCAP file name does not contain any Call ids.
Event attributes	Comma separated key=value attributes to be added in the 'message-log' event generated by this action

Log message

Generates a syslog message.

Parameter	Description
log level	Syslog severity level.
message text	Text of the syslog message.

Log event

Generates a notification.

Parameter	Description
event text	Message text of the notification event.

Traffic Shaping Actions

Sets a quota on SIP and RTP traffic and reports violations.

Limit parallel calls

Sets a quota on number of parallel calls.
The quota can be applied to a specific part of the traffic by using a key.

New calls arriving in excess of this limit are declined using the 403 SIP response.

The limit applies separately to inbound and outbound traffic respectively going through inbound and outbound rules, unless 'global key' is selected, in which case the limit will apply to both inbound and outbound traffic.

Parameter	Description
Parallel calls	Maximum number of parallel calls.
Key attribute	Optional key that identifies a subset traffic.
Is global key	• Checked: The limit applies to inbound and outbound traffic, on all Call Agents for which the key attribute matches. • Unchecked: The limit only applies to the traffic matching this rule and key attribute.
SIP Response code	The response code issued when calls are refused
SIP Response reason	The response reason in the response when calls are refused
SIP header	SIP header to add when call limit is reached. Ie: 'Warning: call limit reached'.
Soft limit	Generate a 'limit' event everytime the counter is greater than this value. Value of 0 disables the soft-limit.
Report abuse	Generate an abuse report for firewall blacklisting.

Limit CAPS

Sets a quota on number of call attempts within a certain time period.
The quota can be applied to a specific part of the traffic by using a key.

New calls arriving in excess of this limit are declined using the 403 SIP response.

The limit applies separately to inbound and outbound traffic respectively going through inbound and outbound rules, unless 'global key' is selected, in which case the limit will apply to both inbound and outbound traffic.

If a SIP call is challenged and a re-INVITE is sent with the same SIP alias, this is considered as being part of the same call and only increases CAPS by one.

Parameter	Description
Call Attemps per time unit	Maximum number of calls within the time period.
Time unit	Period of time (in seconds).
Key attribute	Optional key that identifies a subset traffic.
Is global key	• Checked: The limit applies to inbound and outbound traffic, on all Call Agents for which the key attribute matches. • Unchecked: The limit only applies to the traffic matching this rule and key attribute.
SIP Response code	SIP response code to use when the call limit is reached.
SIP Response reason	The response reason in the response when the call limit is reached.
SIP header	SIP header to add when call limit is reached. Ie: 'Warning: call limit reached'.
Soft limit	Generate a 'limit' event everytime the counter is greater than this value. Value of 0 disables the soft-limit.
Report abuse	Generate an abuse report for firewall blacklisting.

Limit Bandwidth

Sets a quota on RTP traffic bandwidth.
The quota can be applied to a specific part of the traffic by using a key.

New calls arriving in excess of this limit are declined using the 403 SIP response.

The limit applies separately to inbound and outbound traffic respectively going through inbound and outbound rules, unless 'global key' is selected, in which case the limit will apply to both inbound and outbound traffic.

Parameter	Description
Limit Bandwidth	The limit of bandwidth, in kbps, for the traffic matching this rule.
Key attribute	Optional key that identifies a subset traffic.
Is global key	• Checked: The limit applies to inbound and outbound traffic, on all Call Agents for which the key attribute matches. • Unchecked: The limit only applies to the traffic matching this rule and key attribute.
SIP Response code	The response code issued when calls are refused
SIP Response reason	The response reason in the response when calls are refused
SIP header	SIP header to add when the bandwidth limit is reached. Ie: 'Warning: bandwidth limit reached'.
Soft limit	Generate a 'limit' event everytime the counter is greater than this value. Value of 0 disables the soft-limit.
Report abuse	Generate an abuse report for firewall blacklisting.

Set call Timer

Disconnects a call if it exceeds a specified duration.

Parameter	Description
Call Timer	Set the maximum call duration, in seconds.

Media Processing Actions

Media Processing Actions.

Enable RTP Anchoring

Anchors RTP media to the SBC and centralises media forwarding. 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 disconnected.

Parameter	Description
Force symmetric media	When enabled, the SBC ignores the IP address advertised in the SDP payload and learns the IP address and UDP port number of the UA by observing RTP packets coming from the UA. It then starts sending the reverse RTP stream to that address.
Lock on addresses learned from RTP	When enabled, the SBC will not overwrite remote addresses learned fromRTP with addresses learned from signaling.
Enable intelligent relay (IR)	When enabled, the SBC detects that the caller and the callee are behind the same NAT and if so, bypasses the media relay. The test is done by comparing the source IP address of the incoming INVITE message with the intended destination of the request.
Source IP Header field for IR	When intelligent relay is enabled, this value configures a header field used to transport the information about the caller's network through additional proxies in the signaling path.
Offer ICE-lite	When enabled, the SBC adds the ICE-lite capabilities to the SDP offer.
Offer RTCP feedback	Adds additional RTCP capabilities for better QoS monitoring than the one available in traditional RTP implementations. When the option is enabled, the RTP profile is changed from AVP/SAVP to AVPF/SAVPF. When the option is disabled, the RTP profile is changed from AVPF/SAVPF to AVP/SAVP.
Offer RTCP Multiplexing	Multiplexing RTP and RTCP packets on a single port, negotiated as per RFC 5761. When enabled, offers sent will contain the "rtcp-mux" attribute in the SDP. Upon reception of an answer containing the "rtcp-mux" attribute, the RTCP packets will be relayed with the RTP packets through the same port. Regardless of the value of this option, whenever an offer is received with the "rtcp-mux" attribute, the answer sent will also contain the "rtcp-mux" attribute, and the RTCP packets will be relayed with the RTP packets through the same port.
Keepalive	When set to a time value, it allows the SBC to send keep-alive RTP traffic. This is useful if one side of a call detects and discontinues inactive calls whereas the other side suppresses RTP due to Voice Inactivity Detection or On Hold scenarios.
Keepalive method	Selects a format of RTP packets to be used as keep-alive RTP traffic.
Timeout	When set to a time value, allows the SBC to disconnect a call when no RTP traffic appears. This is useful to eliminate "hanging calls" due to abruptly disconnected SIP devices.

Parameter	Description
Change SSRC	When enabled, generates a new, non-zero, SSRC value for the RTP stream going towards the Call Agent peer. This option must be enabled on both inbound and outbound rules in order to convert the streams for both inbound and outbound calls.

SRTP Preferences

Configures SRTP Preferences.

Parameter	Description
Crypto Context Behavior	Specifies whether or not the cryptographic context, more specifically the rollover counter (ROC), must be reset after a SDP renegotiation. The ROC is an internal counter that tracks how many times the sequence number went over 65535. Both endpoints must have an identical behavior, else this will lead to decryption problems. The possible behaviors are: • ResetAlways: Reset the cryptographic context on every renegotiation, even if the crypto key has not changed. • ResetOnNewCrypto: Reset the cryptographic context when the negotiated crypto key has changed. • ResetNever: The cryptographic context is never reset after a renegotiation. The context is updated with the new crypto key and the ROC keeps its current value. Note: When the cryptographic context is reset, the sequence number, timestamp and SSRC of the stream are also reset to new randomized values. Note: In any case, the cryptographic context is automatically reset when the stream destination changes, regardless of the config of this parameter. Note: When this action is not used, the SBC default behavior is 'ResetNever'. Refer to the **Crypto Mode When Sending Offer** and **Crypto Mode When Sending Answer** parameters for more details how cryptographic attributes are negotiated.

Parameter	Description
Crypto Mode When Sending Answer	Specifies if the cryptographic attribute in the SDP answer with secured RTP (SRTP) streams should be regenerated or not. A cryptographic attribute includes the cryptographic suite, the key parameters, and the session parameters. The possible values are: • RegenerateAlways: Always regenerate new cryptographic attribute for the SDP answer to be sent, regardless of the content of the received SDP offer. • KeepUnlessCryptoChanged: Keep the same cryptographic attribute for the SDP answer to be sent, unless the matching cryptographic attribute of the received SDP offer changed. • KeepAlways: Keep the same cryptographic attribute for the SDP answer to be sent, regardless of the content of the received SDP offer When the sent or received cryptographic attribute change, the behavior of the associated cryptographic context can be adjusted according to the **Crypto Context On Change** parameter. This parameter is effective only when the **Key Exchange Mechanisms** is *SDES*. Note: When this action is not used, the SBC default behavior is 'KeepAlways'.

Parameter	Description
Crypto Mode When Sending Offer	Specifies if the cryptographic attribute in the SDP offer should be regenerated or not for a new SDP offer with secured RTP (SRTP) streams. A cryptographic attribute includes the cryptographic suite, the key parameters, and the session parameters. An SDP offer is sent during the following call scenarios: • Hold and Resume • Session timers (i.e. session expires) • Forking • Call Transfer • Conference call Of this list is excluded establisment or termination of a call. The possible values are: • RegenerateAlways: Always regenerate new cryptographic attribute for the SDP offer to be sent. • KeepUnlessMediaChanged: Keep the same cryptographic attribute for the SDP offer to be sent, unless the media stream state changed (e.g. a resume or call transfer situation). • KeepAlways: Keep the same cryptographic attribute for the SDP offer to be sent. When the sent or received cryptographic attribute changes, the behavior of the associated cryptographic context can be adjusted according to the **Crypto Context On Change** parameter. This parameter is effective only when the **Key Exchange Mechanisms** is *SDES*. Note: When this action is not used, the SBC default behavior is 'KeepAlways'.

Parameter	Description
SRTP lifetime	When selected, the lifetime is appended at the end of the cryptographic attribute.
Support AES_256_CM_HMAC_SHA1_80 Crypto Suite	When selected, the AES_256_CM_HMAC_SHA1_80 Crypto Suite is considered supported. Note: At least one suite must be selected to allow the communication to be established.
Support AES_CM_128_HMAC_SHA1_80 Crypto Suite	When selected, the AES_CM_128_HMAC_SHA1_80 Crypto Suite is considered supported. Note: At least one suite must be selected to allow the communication to be established.
Support AES_CM_128_HMAC_SHA1_32 Crypto Suite	When selected, the AES_CM_128_HMAC_SHA1_32 Crypto Suite is considered supported. Note: At least one suite must be selected to allow the communication to be established.

Force RTP/SRTP

Force the media traffic to use secure or non-secure RTP.
This action has a different effect depending on which call leg it is used.

- When used in an inbound rule, forces the caller to offer RTP/SRTP. This action does not require RTP anchoring.
- When used in an outbound rule, converts the media packet to use RTP/SRTP. RTP anchoring must be enabled in an inbound or an outbound rule otherwise this action is ignored.

Parameter	Description
Force plain RTP	When selected, forces/converts the side of the call it is used on to RTP. Selecting it disables "Force secure RTP".
Force secure RTP	When selected, forces/converts the side of the call it is used on to SRTP. Selecting it disables "Force plain RTP".
Key Exchange Mechanisms	Only usable when "Force secure RTP" is activated. Offers the use of DTLS, SDES or both to secure the key exchange mechanism.

DTMF conversion from INFO to avt

Convert DTMF from SIP INFO request into RTP AVT format according to RFC 2833.
This action is sensitive to the place where it is used:

- As an inbound rule on the caller side, it converts the DTMF sent by the caller.
- As an outbound rule on the callee side, it converts the DTMF sent by the callee.
- As an inbound rule on the callee side, or as an outbound rule on the caller side, it has no effect.

Example: Call Agent A handles peers that use SIP INFO, and Call Agent B handles peers that use RTP AVT. We want DTMF converted between Call Agents A and B in both directions. The following configuration works:

- Action dtmf_info2avt must be used as inbound and outbound rules on Call Agent A.
- Action dtmf_avt2info must be used as inbound and outbound rules on Call Agent B.

DTMF conversion from avt to INFO

Convert DTMF from RTP AVT format (RFC 2833) into SIP INFO request.
This action is sensitive to the place where it is used:

- As an inbound rule on the caller side, it converts the DTMF sent by the caller.
- As an outbound rule on the callee side, it converts the DTMF sent by the callee.
- As an inbound rule on the callee side, or as an outbound rule on the caller side, it has no effect.

Example: Call Agent A handles peers that use SIP INFO, and Call Agent B handles peers that use RTP AVT. We want DTMF converted between Call Agents A and B in both directions. The following configuration works:

- Action dtmf_info2avt must be used as inbound and outbound rules on Call Agent A.
- Action dtmf_avt2info must be used as inbound and outbound rules on Call Agent B.

SIP dropping Actions

Eliminating non-compliant traffic, either silently or with a SIP response.

Reply to request with reason and code

Sends a response to a SIP request.
When this action is executed, subsequent actions and rules are not executed.

Parameter	Description
Code	SIP error code
Reason phrase	Reason text Replacement expressions can be part of this value.
Optional header field	Header fields to include in the response message. Replacement expressions can be part of this value. Multiple headers can be added by inserting '\r\n' between them.

Allow unsolicited NOTIFYs

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

Drop request

Drops requests silently.
When this action is executed, subsequent actions and rules are not executed.

Parameter	Description
Event throttling key	A replacement expression. When a request is dropped, an event is generated for each different value of the replacement expression. For example: If the event throttling key is set to '$si', an event will be generated once for each different source IP address of the dropped request. When empty, an event is generated for each dropped request. When the key contains a constant string, an event is generated at most once every 10 seconds.
Blacklist by firewall if repeated	Include this action when considering blacklisting IP address in firewall.
Drop reason text	Drop reason text.

Scripting Actions

Processing of internal parameters that are used to link multiple actions together using intermediate results stored in parameters.

Set Call parameter

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

Parameter	Description
parameter name	Name of the parameter to set.
parameter value	Value of the parameter.Replacement expressions can be part of this value.

Registration Processing Actions

REGISTER caching, storing and throttling.

Enable REGISTER caching

Generates an alias for the contact URI of a REGISTER request and forwards the request with the alias instead of the original contact.
Upon successful answer to the request, the alias is stored together with the original contact and AOR in the registration cache.

Parameter	Description
Cookie method	Determines where the alias is located in the Contact header field. • User part: The contact is generated in the "<sip:[alias]@[SBC IP and port]>." form • reg_alias URI parameter: The contact is generated in the "<sip:[SBC IP and port];reg_alias=[alias]>" form. • Username (no alias): The contact username is used directly as the "cookie", and is not replaced by an "alias". This method has the limitation of not supporting multiple registrations per user. Only the last registration is saved in the cache for any given user. This method should only be used when the server replies without an alias in the SIP REGISTER responses.
With temporary aliases	This option will create temporary aliases when the REGISTER request is relayed so that requests sent by the registrar shortly after the REG/200 can be relayed to the registrering UA even if this request is processed before the 200/REG.

Retarget R-URI from cache

Lookup the registration cache to find an entry with an alias matching the user-part of the request’s R-URI.
If a matching entry is found, the R-URI of the request is re-written with the original contact from the registration cache.

If no entry is found, the request is not modified and is processed with the next applicable rules.

Note that entries with aliases are saved in the registration cache when using 'Enable REGISTER caching'.

Parameter	Description
enable NAT handling	When enabled, the SBC sends subsequent SIP messages to the source IP and port of the REGISTERS request it received.
enable sticky transport	When enabled, the SBC uses the same interface and transport over which the REGISTER was received for sending subsequent SIP messages.

REGISTER throttling

Forces SIP user-agents to shorten the re-registration period while propagating the REGISTERs upstream to the registrar, at longer intervals.
Particularly useful to trigger REGISTER-based keep-alives to facilitate NAT traversal.

This action requires the use of "enable REGISTER caching" or "save REGISTER contact" actions and must precede any REGISTER processing action in order to take effect.

This action is executed immediately when the rule is evaluated. The reply is sent immediately.

The order of the rules is important. The rules and the actions are executed in the order in which they appear.

Registration requests are forwarded for UAS re-registration when the binding expires at the UAS or when the source IP/Port has changed.

Parameter	Description
minimum registrar expiration	Sets the minimum interval between registrations of the SBC unit to the registrar. This value is assigned to the **expires** parameter in the REGISTER sent to the registrar when the **expires** parameter in the REGISTER received from the user-agent is smaller or when the parameter is absent.
maximum UA expiration	Sets the maximum interval between registrations of the registering user to the SBC unit. This value is assigned to the **expires** parameter in the response sent to the user-agent when the **expires** parameter in the REGISTER received from the user-agent is larger or when the parameter is absent. This value must be smaller than the registrar expiration or at least twice as big.

Save REGISTER contact

Acts as a local registrar and stores the AOR and contact in the registration cache.
This contact overwrites any previously saved contact.

The REGISTER request is answered with a 200 OK and no further processing is done on that request.

Restore contact from registrar

Lookup the registration cache to find an entry with an AOR matching the request's request URI.
If a matching entry is found, the request URI of the request is re-written with the original contact from the registration cache.

If no entry is found, the request is not modified and is processed with the next applicable rules.

Note that this action will restore contacts that were saved using 'Save REGISTER contact' or 'Enable REGISTER caching'.

Fetch home-proxy IP

This action retrieves the routing information used for user registration and routes later requests to the user's 'home proxy', without re-targeting DNS lookup. This is useful in IMS use-cases when the user's signaling must be routed through the proxy previously used for registration.
This action sets an internal 'next-hop' value available for a routing rule that uses the 'next hop' routing method. This internal 'next-hop' value is set to the IP address, port and outbound interface to reach the proxy where the user was previously registered.

Please note that for this action to work, the 'source-alias' call parameter must be set. This can be done by a previous Register Cache condition targeting the caller (e.g. Register Cache / From URI (AoR +Contact+IP/port) / 'Is Registered').

External Interaction Actions

Queries to external servers by REST or ENUM.

ENUM query

Query an ENUM server and replaces the R-URI with the result of the query. If no source is entered, the R-URI user is used. The default domain suffix can also be overridden to query private ENUM servers or non E164 numbers.

Parameter	Description
queried value	Field to be queried by the Enum query.
domain suffix	DNS suffix used for the Enum query (ex: e164.arpa).
ENUM services	Semi-colon separated list of ENUM services (ex: sip;voice:sip;video:sip).

Read call parameters over REST

Do REST query to given URL and set call variables received in reply. Typically input to the query is specified as URI parameters that include substitution variables.

Parameter	Description
REST URL	REST server URL

NAT Handling Actions

Facilitates NAT traversal.

Enable dialog NAT handling

Remembers during the dialog duration where the initial dialog initiating request came from and sends all subsequent traffic to that IP address and port.

Other Actions

Miscellaneous Call Agent actions.

Fork

Duplicates a SIP request. When forking is used, extra sessions are required for the additional call legs involved. Make sure your licences support enough sessions.

Parameter	Description
New R-URI	Requests the URI of the new request.

Add Q.850 Reason Header

Add a Reason header with the Q.850 call termination reason, if no Reason header is present.

Parameter	Description
on inbound leg	Add the header on the inbound call leg.
on outbound leg	Add the header on the outbound call leg.

Reset dialog by 1xx replies

Allow early dialog to be reset when a downstream serial-forking proxy forwards a call to another destination without 181 reply and previous 1xx replies were not important (without SDP, not reliable).

SBC Routing Rule Actions

Routing Actions

Static and dynamic SIP request routing.

Prefix-based routing

This is frequently used when you have a number of PSTN gateways serving different regions. Technically you match area codes with the beginning of the user-part of the request URI.

Parameter	Description
Match	This parameter specifies an expression on which the prefix matching is done. This expression can be made of Replacement expressions, call parameters, backreferences to a regular expression evaluated in the rule condition, ... For instance, setting this parameter to $rU causes the match to be checked against the user part of the request URI. The prefix matching searches for the corresponding entry in the **PrefixBasedRouting** table. Then the routing is done according to the parameters in the matching entry in the **PrefixBasedRouting** table.
Update To-URI host	Update To-URI host
Replace To-URI host with the resolved destination IP address	Replace To-URI host with the resolved destination IP address

Static route

The Static Route is the simplest type of routing where the administrator explicitly chooses the destination Call Agent.
The next SIP hop is determined by either using the request URI or by directly using an IP address.

Whichever method is chosen, the Rulesets of the specified destination Call Agent are executed.

If the next hop yields multiple IP addresses (e.g. when using an FQDN), the SBC load-balances among them by their respective priorities.

Parameter	Description
Call Agent	The Call Agent to which the request will be forwarded.
Route via R-URI	The 'Route via R-URI' method uses the request URI to find out the next-hop IP address. That is particularly useful when Call Agent rules alter the request URI using actions like reverse registration cache or ENUM lookup. If the host part of the request URI includes a DNS name that resolves to multiple destinations per RFC 3263, the SBC load-balances among the respective destinations according to their priorities.
Set Next Hop	Uses the destination Call Agent's peer IP address(es) as the next-hop IP address. Optional sub-parameters: • Use another destination instead of Call Agent's destination(s): Overrides the CA's peer IP address by the provided IP address. • Use on first request only: This option changes the default behaviour for forwarding subsequent in-dialog requests. By default when turned off, all subsequent outbound requests will follow the path of the dialog-initiating request. However, if this option is turned on, the next-hop logic for subsequent requests is governed only by the SIP standard procedures. Particularly, if the next hop in the INVITE path was a non-record-routing proxy, it will not be included in the request path. • Update R-URI host: This option rewrites the host part of the request URI with the address of the next hop. By default it is turned off and the request URI is not modified when forwarded. • Add Route header field: This option is also known as "preloaded Route". It prints the next-hop destination in the Route header field. Use only if next SIP hop is known to require such a behaviour.
Replaces the DNS name in the request URI with the resolved IP address.	If a DNS name appears in the request URI, it is substituted by its resolved IP address before forwarding.
Update To-URI host	Update To-URI host

Parameter	Description
Replace To-URI host with the resolved destination IP address.	Replace To-URI host with the resolved destination IP address.
Force transport	Overrides the transport protocol to be used to initiate SIP dialogs with the next hop. The following protocols are supported: UDP, TCP, TLS and Websockets. Note: To force TLS and guarantee that no other transport protocol can be used, you must also set the **SignalingInterface.AllowedTransports** parameter to TlsOnly.
Enable redirect handing	If this option is enabled, incoming 302 SIP responses are not passed upstream. Instead, the SBC takes the content of the 'Contact' header field and uses it as another next-hop for forwarding the original request. Specifically, the contact URI in the 302 SIP response is used to rewrite the request URI, determine the next-hop IP address and looks up a Call Agent whose outbound rules are processed.

Call Agent based on R-URI

The SBC tries to find a Call Agent that matches the host in the request URI.
This can be particularly useful if a rule changes the hostname in a request URI, for example by ENUM lookup. If the lookup yields an address of a valid Call Agent, it is used for routing, otherwise it proceeds to the next rule.

Parameter	Description
Route via R-URI	The 'Route via R-URI' method uses the request URI to find out the next-hop IP address. That is particularly useful when Call Agent rules alter the request URI using actions such as 'Restore contact from registrar' or 'ENUM lookup'. If the host part of request URI includes a DNS name that resolves to multiple destinations per RFC 3263, the SBC load-balances among the respective destinations by their priorities.
Set Next Hop	Uses the destination Call Agent's peer IP address(es) as the next-hop IP address.
Replaces the DNS name in the request URI with the resolved IP address.	If a DNS name appears in the request URI, it is substituted by its resolved IP address before forwarding.
Update To-URI host	Update To-URI host
Replace To-URI host with the resolved destination IP address.	Replace To-URI host with the resolved destination IP address.
Force transport	Overrides the transport protocol to be used to initiate SIP dialogs with the next hop. The following protocols are supported: UDP, TCP, TLS and Websockets. Note: To force TLS and guarantee that no other transport protocol can be used, you must also set the **SignalingInterface.AllowedTransports** parameter to TlsOnly.
Enable redirect handing	If this option is enabled, incoming 302 SIP responses are not passed upstream. Instead, the SBC takes the content of the 'Contact' header field and uses it as another next-hop for forwarding the original request. Specifically, the contact URI in the 302 SIP response is used to rewrite the request URI, determine the next-hop IP address and look up a Call Agent whose outbound rules are processed.

SBC Replacement Expressions

$r

Request URI.
The expression refers to the current request URI which may be changed during the course of request processing.

Name	Description
$r.	Complete request URI.
$ru	user@host[:port] part of the request URI.
$rU	User part of the request URI.
$rd	Request-URI domain (host:port).
$rh	Host part of the request URI.
$rp	Port number of the request URI.
$rP	Request URI Parameters.

$f

From header

Name	Description
$f.	Complete 'From' header field.
$fu	user@host[:port] part of the 'From' URI.
$fU	User part of the 'From' URI.
$fd	'From' URI domain (host:port).
$fh	Host part of the 'From' URI.
$fp	Port number of the 'From' URI.
$fn	Display name of the 'From' header field.
$fP	URI parameters of the 'From' header field.
$fQ	Header parameters of the 'From' header field.
$ft	The tag of the 'From' header field.
$fH	The name of the 'From' header field, exactly as it is stated in the SIP request.

$t

To header

Name	Description
$t.	Complete 'To' header.
$tu	user@host[:port] part of the 'To' URI.
$tU	User part of the 'To' URI.
$td	'To' URI domain (host:port).
$th	Host part of the 'To' URI.
$tp	Port number of the 'To' URI.
$tn	Display name of the 'To' header field.
$tP	URI parameters of the 'To' header field.
$tQ	Header parameters of the 'To' header field.
$tt	The tag of the 'To' header.
$tH	The name of the 'To' header, exactly as it is stated in the SIP request.

$a

P-Asserted-Identity

Name	Description
$a.	Complete 'P-Asserted-Identity' header.
$au	user@host[:port] part of the 'P-Asserted-Identity' URI.
$aU	User part of the 'P-Asserted-Identity' URI.
$ad	'P-Asserted-Identity' URI Domain (host:port).
$ah	Host part of the 'P-Asserted-Identity' URI.
$ap	Port number of the 'P-Asserted-Identity' URI.
$aP	Parameters of the 'P-Asserted-Identity' header field. Does not include the parameters of the URI.
$at	The tag of the 'P-Asserted-Identity'.
$aH	The name of the 'P-Asserted-Identity' header field, exactly as it is stated in the SIP request.

$p

P-Preferred-Identity

Name	Description
$p.	Complete 'P-Preferred-Identity' header field.
$pu	user@host[:port] part of the 'P-Preferred-Identity' URI.
$pU	User part of the 'P-Preferred-Identity' URI.
$pd	'P-Preferred-Identity' URI Domain (host:port).
$ph	Host part of the 'P-Preferred-Identity' URI.
$pp	Port number of the 'P-Preferred-Identity' URI.
$pP	Parameters of the 'P-Preferred-Identity' header field. Does not include the parameters of the URI.
$pt	The tag of the 'P-Preferred-Identity' header field.
$pH	The name of the 'P-Preferred-Identity' header, exactly as it is stated in the SIP request.

$c

Call-ID

Name	Description
$ci	Call-ID of the SIP request.

$s

Source party

Name	Description
$si	Source IP address of the inbound SIP request.
$sp	Source port number of the inbound SIP request.

$d

Expected destination party

Name	Description
$di	Destination IP address of the outbound SIP request. This replacement expression is only available in outbound rules.
$dp	Destination port number of the outbound SIP request. This replacement expression is only available in outbound rules.

$R

Interface of the inbound SIP request

Name	Description
$Ri	IP address of the Signaling Interface on which the inbound SIP request was received.
$Rp	Port number of the Signaling Interface on which the inbound SIP request was received.
$Rf	ID of the Signaling Interface on which the inbound SIP request was received.
$Rn	Name of the Signaling Interface on which the inbound SIP request was received.
$Rl	The configured address of the **SignalingInterface.PublicIpAddr** parameter on which the inbound SIP request was received.

$H

Arbitrary Headers
The replacement expressions in this group mention the name of an arbitrary header between parentheses.

The core headers (From, To, Call-ID, Via, Route, Record-Route, Contact) cannot be replaced.

Example: $H(Server) will be replaced by the value of the 'Server' header field.

Name	Description
$H(headername)	Value of the 'headername' header
$HU(headername)	User part of the URI in the 'headername' header.
$Hd(headername)	URI Domain (host:port) of the 'headername' header.
$Hu(headername)	user@host[:port] part of the URI in the 'headername' header.
$Hh(headername)	Host part of the URI in the 'headername' header.
$Hp(headername)	Port number of the URI in the 'headername' header.
$Hn(headername)	Display name of the 'headername' header.
$HP(headername)	Parameters of the 'headername' header field. Does not include the parameters of the URI.
$HH(headername)	Header headername (as URI) headers

$m

Request method

Name	Description
$m	The method of the request.

$V

Call parameter

Name	Description
$V(gui.varname)	Value of the 'varname' call parameter.

$B

Cnum and Rnum

Name	Description
$B(cnum.rnum)	Value of the regular expression backreference. • Rnum is the index of the rule condition containing the regular expression, starting at 1. • Cnum is the index of the subexpression within the regular expression, starting at 1.

$U

Register cache

Name	Description
$Ua	Originating AoR from the registration cache. This replacement expression can only be used after the execution of a‘Restore contact from registrar’ or a ‘Retarget R-URI from cache’ action.
$UA	Originating alias from the registration cache. This replacement expression can only be used after the execution of a ‘Restore contact from registrar’ or a ‘Retarget R-URI from cache’ action.

$_

Operations on Values

Name	Description
$_u(value)	Changes the value to uppercase.
$_l(value)	Changes the value to lowercase.
$_s(value)	Size of the value.
$_5(value)	MD5 sum of the value.
$_r(value)	Random number from 0 to 'value' - 1. Example: $_r(5) gives 0, 1, 2, 3 or 4.

$#

URL-encoded

Name	Description
$#(value)	Encodes the value into a URL format with appropriate escaping for characters outside the ASCII character set.

$attr

Global Attributes

Name	Description
$attr(version)	Version number of the DGW firmware. The returned value matches with the %version% macro in DGW firmware.
$attr(profile)	Profile identification of the DGW firmware. The returned value matches with the %profile% macro in DGW firmware.
$attr(serial)	Serial number of the Mediatrix unit. The returned value matches with the %serial% macro in DGW firmware.
$attr(mac)	MAC address of the Mediatrix unit. The returned value matches with the %mac% macro in DGW firmware.
$attr(product)	Product name of the Mediatrix unit. The returned value matches with the %product% macro in DGW firmware.
$attr(productseries)	Product series name of the Mediatrix unit. The returned value matches with the %productseries% macro in DGW firmware.

Copyright Notice

media5 corporation

media5corp.com