Brekeke SIP Server

Version 2.0

Dial Plan Tutorial

Brekeke Software, Inc.

Version

Brekeke SIP Server v2.0 Dial Plan Tutorial

Revised: January 12, 2007

Copyright

Disclaimer

Brekeke Software, Inc. reserves the right to change any information found in this document without any written notice to the user.

Trademark Acknowledgement

- *LINUX is a registered trademark of Linus Torvalds in the United States and other countries.*
- *Red Hat is a registered trademark of Red Hat Software, Inc.*
- *Windows is a trademark or registered trademark of Microsoft Corporation in the United States and other countries.*
- *Mac is a trademark of Apple Computer, Inc., registered in the U.S. and other countries.*
- *Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.*
- *Other logos and product and service names contained in this document are the property of their respective owners.*

1. INTRODUCTION 6

2. ROUTING 6

2.1. Routing Setting by the Destination SIP URI 7

Ex 1 Routing all calls to sip:user@host 7

Ex 2 Routing a call to sip:user@host if the callee’s name is "admin" 7

Ex 3 Routing a call to sip:user@host if the callee’s SIP URI is sip:admin@server 7

Ex 4 Routing a call to “host” with the same username if the callee's name prefix is "9" 7

Ex 5 Routing a call to "host" without the prefix if the callee's name prefix is "9" 8

Ex 6 Routing a call to "host" with the prefix "8" if the callee's name prefix is "9" 8

Ex 7 Routing a call to sip:user@host if the callee isn't registered 8

Ex 8 Routing a call to sip:user@host if the caller's name is "admin" 8

Ex 9 Routing a call to sip:user@host if the caller isn't registered 8

Ex 10 Routing a call to sip:user@host if the call is from 192.168.0.1 8

2.2. Routing Setting by the Destination User Name 9

Ex 11 Routing a call to the user "user" if the user is registered 9

Ex 12 Routing a call to the user "user" if the callee isn't registered 9

Ex 13 Routing a call to the user who was registered as the callee's name with the prefix “9” 9

Ex 14 Routing a call to the user "user" from 10:00AM to 5:59PM 9

Ex 15 Routing a call to the user "user" from December 12 to December 19 9

2.3. Routing Setting by the Destination IP Address or FQDN 10

Ex 16 Routing a call to "server" if the callee's host name is "host" 10

Ex 17 Routing a call to "host.domain" if the call is from 192.168.0.1 10

Ex 18 Routing a call to "host.domain" if the call is from the localhost 10

Ex 19 Routing a call to 192.168.0.100 if the call is from the port number 15060 10

Ex 20 Routing a call to 192.168.0.100 if the request method is SUBSCRIBE 10

3. REJECTING 11

Ex 21 Returning a "603 Decline" response if the callee isn't registered 11

Ex 22 Returning a "486 Busy" response if the callee's SIP URI is sip:user@host 11

Ex 23 Returning a "402 Payment Required" response if the callee's name prefix is "9" 11

Ex 24 Returning a "404 Not Found" response if the caller's name is "user" 11

Ex 25 Returning a "403 Forbidden" response if the call is from an IP address with the prefix "192.168" 12

Ex 26 Returning a "406 Not Acceptable" response if the Content-Type header is "application/text". 12

Ex 27 Returning a "503 Service Unavailable" response if the User-Agent header contains "TEST" ... 12
Ex 28 Returning a "483 Too Many Hops" response if the Max-Forwards' value is 5 or less 12
Ex 29 Returning a "480 Temporarily Unavailable" response from 0:00AM to 7:59AM 12
Ex 30 Returning a "400 Bad Request" response if the request method is SUBSCRIBE 12

4. EDITING SIP HEADERS 13

4.1. Replacing an Existing SIP Header 13
Ex 31 Changing the caller's display name to "Ted" if his/her user name is "admin" 13
Ex 32 Changing the Expires's value to 200 if it is less than 200 13
Ex 33 Replacing the User-Agent's value to contain "Beta" if it contains "Alpha" 13

4.2. Appending SIP Header 14
Ex 34 Appending new header "X-Example" 14

4.3. Deleting SIP Header 14
Ex 35 Deleting the User-Agent header 14

5. AUTHENTICATION 14

Ex 36 Requiring Authentication if the callee's domain name is "host.domain" 14
Ex 37 Not Requiring Authentication if the callee's name prefix is "800" 15
Ex 38 Requiring Authentication if the caller isn't registered 15
Ex 39 Requiring Authentication if the call is from an IP address with the prefix "192.168.10" 15
Ex 40 Not Requiring Authentication from 10:00AM to 5:59PM 15

6. LOAD BALANCING 16

Ex 41 Load Balancing by switching 3 destinations every second 16
Ex 42 Load Balancing by switching 2 destinations every 30 minutes 16
Ex 43 Load Balancing based on whether the Session ID is odd or even 16

7. NAT TRAVERSAL 17

7.1. Setting NAT Traversal ON/OFF 17
Ex 44 Enabling NAT Traversal if the callee's domain name is "host.domain" 17
Ex 45 Disabling NAT Traversal if the call is from 192.168.0.1 17

7.2. Specifying the Interface Address 18
Ex 46 Using "192.168.1.1" as the interface address if the prefix of callee's contact address is "192.168.1" 18
Ex 47 Using "192.168.2.1" as the interface address if the call is from an IP address with the prefix "192.168.2" 18

8. RTP RELAY 19

Ex 48 Enabling RTP Relay if the callee's name prefix is "9" 19
Ex 49 Enabling RTP Relay and using PCMA as the codec if the callee's SIP URI is sip:user@host .. 19
Ex 50 Enabling RTP Relay and assigning the range of ports from 10000 to 10100 if the is call from 192.168.0.1 20

9. SPECIFYING ENVIRONMENT VARIABLES........20

Ex 51 Using G723 as the codec for all calls 20
Ex 52 Not Appending Record-Route header if the callee's name prefix is "9" 20
Ex 53 Not Adding lr parameter to Record-Route header if the callee's host name is "host" 20
Ex 54 Not Appending rport parameter to Via header if the callee's host name is "host" 21
Ex 55 Setting the ringing timeout period to 30 seconds if the caller's name is "admin" 21
Ex 56 Using Upper Registration to "host.domain" if the caller's name prefix is "9" 21
Ex 57 Adjusting the following registration period as 100 seconds if the current period is less than 100 seconds 21
Ex 58 Not Using Thru Registration if the callee's host name is "host" 21

10. USING SESSION PLUG-IN 22

Ex 59 Using "RadiusAcct" plug-in for all calls 22
Ex 60 Using "CDRlog" plug-in if the callee's host name is "host" 22

1. Introduction

This document introduces various samples of Brekeke SIP Server Dial Plan rules. For the basics of Dial Plan, syntaxes, and how to set dial plan rules using the Brekeke SIP Server Admintool, refer to the "Brekeke SIP Server Administrator's Guide, Section 6. Dial Plan".

The Dial Plan features explained in this document are as follows:

- Routing
- Rejecting
- Editing SIP Headers
- Authentication
- Load Balancing
- NAT Traversal
- RTP Relay
- Specifying Environment Variables
- Using Session Plug-in

2. Routing

Routing is the major feature of Dial Plan. There are three ways to define routing using Deploy Patterns. The routing setting will be enabled only when the corresponding conditions in Matching Patterns are fulfilled.

To = destination SIP URI

Example: To = sip:user@host

The session will be routed to the "host".

To = destination user name

Example: To = sip:user@

The session will be routed to the destination user's contact address which was registered in the server's register database when REGISTER request was sent from the user.

$target = destination IP address or FQDN

Example: $target = sip:user@host

$target = host

The session will be routed to the “host”.

Example: $target = 192.168.0.10

The session will be routed to the 192.168.0.10.

2.1. Routing Setting by the Destination SIP URI

Ex 1 Routing all calls to sip:user@host

Matching Patterns	Deploy Patterns
`$request = ^INVITE`	`To = sip:user@host`

Ex 2 Routing a call to sip:user@host if the callee’s name is "admin"

Matching Patterns	Deploy Patterns
`$request = ^INVITE` `To = sip:admin@`	`To = sip:user@host`

Ex 3 Routing a call to sip:user@host if the callee’s SIP URI is sip:admin@server

Matching Patterns	Deploy Patterns
`$request = ^INVITE` `$geturi(To) = sip:admin@server`	`To = sip:user@host`

Ex 4 Routing a call to “host” with the same username if the callee's name prefix is "9"

Matching Patterns	Deploy Patterns
`$request = ^INVITE` `To = sip:(9.+)@`	`To = sip:%1@host`

Ex 5 Routing a call to "host" without the prefix if the callee's name prefix is "9"

Matching Patterns	Deploy Patterns
`$request = ^INVITE` `To = sip:9(.+)@`	`To = sip:%1@host`

Ex 6 Routing a call to "host" with the prefix "8" if the callee's name prefix is "9"

Matching Patterns	Deploy Patterns
`$request = ^INVITE` `To = sip:(9.+)@`	`To = sip:8%1@host`

Ex 7 Routing a call to sip:user@host if the callee isn't registered

Matching Patterns	Deploy Patterns
`$request = ^INVITE` `$registered = false`	`To = sip:user@host`

Ex 8 Routing a call to sip:user@host if the caller's name is "admin"

Matching Patterns	Deploy Patterns
`$request = ^INVITE` `From = sip:admin@`	`To = sip:user@host`

Ex 9 Routing a call to sip:user@host if the caller isn't registered

Matching Patterns	Deploy Patterns
`$request = ^INVITE` `$registered( From ) = false`	`To = sip:user@host`

Ex 10 Routing a call to sip:user@host if the call is from 192.168.0.1

Matching Patterns	Deploy Patterns
`$request = ^INVITE` `$addr = 192\.168\.0\.1$`	`To = sip:user@host`

2.2. Routing Setting by the Destination User Name

Ex 11 Routing a call to the user "user" if the user is registered

Matching Patterns	Deploy Patterns
`$request = ^INVITE` `$registered( “user” ) = true`	`To = sip:user@`

Ex 12 Routing a call to the user "user" if the callee isn't registered

Matching Patterns	Deploy Patterns

`$request = ^INVITE`
`$registered = false`
`$registered( “user” ) = true`
	`To = sip:user@`

Ex 13 Routing a call to the user who was registered as the callee's name with the prefix “9”

Matching Patterns	Deploy Patterns
`$request = ^INVITE` `To = sip:(.+)@`	`To = sip:9%1@`

Ex 14 Routing a call to the user "user" from 10:00AM to 5:59PM

Matching Patterns	Deploy Patterns
`$request = ^INVITE` `$time =^1[0-7]:`	`To = sip:user@`

Ex 15 Routing a call to the user "user" from December 12 to December 19

Matching Patterns	Deploy Patterns
`$request = ^INVITE` `$date = 12/1[2-9]$`	`To = sip:user@`

2.3. Routing Setting by the Destination IP Address or FQDN

Ex 16 **Routing a call to "server" if the callee's host name is "host"**

Matching Patterns	Deploy Patterns
`$request = ^INVITE` `$geturi( To ) = @host`	`$target = server`

Ex 17 **Routing a call to "host.domain" if the call is from 192.168.0.1**

Matching Patterns	Deploy Patterns
`$request = ^INVITE` `$addr = 192\.168\.0\.1$`	`$target = host.domain`

Ex 18 **Routing a call to "host.domain" if the call is from the localhost**

Matching Patterns	Deploy Patterns
`$request = ^INVITE` `$localhost = true`	`$target = host.domain`

Ex 19 **Routing a call to 192.168.0.100 if the call is from the port number 15060**

Matching Patterns	Deploy Patterns
`$request = ^INVITE` `$port = ^15060$`	`$target = 192.168.0.100`

Ex 20 **Routing a call to 192.168.0.100 if the request method is SUBSCRIBE**

Matching Patterns	Deploy Patterns
`$request = ^SUBSCRIBE`	`$target = 192.168.0.100`

3. Rejecting

Here are some examples of error messages returned for rejecting calls. Error responses specified in the handling variable $action are sent to the request sender. The session for which an error response was returned will not be routed to the call destination.

For the response codes, refer to RFC3261.

Ex 21 Returning a "603 Decline" response if the callee isn't registered

Matching Patterns	Deploy Patterns
`$request = ^INVITE` `$registered = false`	`$action = 603`

Ex 22 Returning a "486 Busy" response if the callee's SIP URI is sip:user@host

Matching Patterns	Deploy Patterns
`$request = ^INVITE` `$geturi( To ) = sip:user@host`	`$action = 486`

Ex 23 Returning a "402 Payment Required" response if the callee's name prefix is "9"

Matching Patterns	Deploy Patterns
`$request = ^INVITE` `To = sip:9.+@`	`$action = 402`

Ex 24 Returning a "404 Not Found" response if the caller's name is "user"

Matching Patterns	Deploy Patterns
`$request = ^INVITE` `From = sip:user@`	`$action = 404`

Ex 25 Returning a "403 Forbidden" response if the call is from an IP address with the prefix "192.168"

Matching Patterns	Deploy Patterns
`$request = ^INVITE` `$addr = 192\.168\.`	`$action = 403`

Ex 26 Returning a "406 Not Acceptable" response if the Content-Type header is "application/text"

Matching Patterns	Deploy Patterns
`$request = ^INVITE` `Content-Type=^application/text$`	`$action = 406`

Ex 27 Returning a "503 Service Unavailable" response if the User-Agent header contains "TEST"

Matching Patterns	Deploy Patterns
`$request = ^INVITE` `User-Agent = TEST`	`$action = 503`

Ex 28 Returning a "483 Too Many Hops" response if the Max-Forwards' value is 5 or less

Matching Patterns	Deploy Patterns
`$request = ^INVITE` `Max-Forwards = ^[0-5]$`	`$action = 483`

Ex 29 Returning a "480 Temporarily Unavailable" response from 0:00AM to 7:59AM

Matching Patterns	Deploy Patterns
`$request = ^INVITE` `$time = ^0[0-7]:`	`$action = 480`

Ex 30 Returning a "400 Bad Request" response if the request method is SUBSCRIBE

Matching Patterns	Deploy Patterns
`$request = ^SUBSCRIBE`	`$action = 400`

4. Editing SIP Headers

Editing a SIP header means replacing the SIP header's contents to a specified value or to add a SIP header or to delete a SIP header. If the specified SIP header field exists in the SIP packet, the contents of the SIP header will be replaced with a new value. If it doesn't exist, the header field will be added to the SIP packet. If the setting value is empty (the text length is 0), the SIP header will be removed from the SIP packet.

By editing a specified SIP header, it is possible to block leaks of the caller's information or to become interoperable with the call destination.

4.1. Replacing an Existing SIP Header

Ex 31 Changing the caller's display name to "Ted" if his/her user name is "admin"

Matching Patterns	Deploy Patterns
`$request = ^INVITE` `$geturi(From) = (sip:admin@.+)`	`From = "Ted" <%1>` `$replaceuri = true`

Ex 32 Changing the Expires's value to 200 if it is less than 200

Matching Patterns	Deploy Patterns
`$request = ^INVITE` `Expires = ^[01][0-9].$`	`Expires = 200`

Ex 33 Replacing the User-Agent's value to contain "Beta" if it contains "Alpha"

Matching Patterns	Deploy Patterns
`$request = ^INVITE` `User-Agent = (.*)Alpha(.*)`	`User-Agent = %1Beta%2`

4.2. Appending SIP Header

Ex 34 Appending new header "X-Example"

Matching Patterns	Deploy Patterns
`$request = ^INVITE`	`X-Example = hello`

4.3. Deleting SIP Header

Ex 35 Deleting the User-Agent header

Matching Patterns	Deploy Patterns
`$request = ^INVITE`	`User-Agent =`

5. Authentication

It is possible to enable authentication only for the specified requests using Dial Plan. Even when authentication is enabled in the System's setting (at the [Configuration] page), it is possible to disable the authentication only for a specified request. The handling variable $auth will be used for this purpose. If its value is true, the authentication will be enabled. If it is false, authentication will be disabled.

If authentication is enabled, Brekeke SIP Server will send the message "407 Proxy Authentication Required（or 401 Unauthorized)" to the request sender and make the sender resend a request with authentication information.
If the request does not include the valid authentication information, Brekeke SIP Server will not authorize the request.

Ex 36 Requiring Authentication if the callee's domain name is "host.domain"

Matching Patterns	Deploy Patterns
`$request = ^INVITE` `$geturi( To ) = @host.domain`	`$auth = true`

Ex 37 Not Requiring Authentication if the callee's name prefix is "800"

Matching Patterns	Deploy Patterns
`$request = ^INVITE` `To = sip:800.+@`	`$auth = false`

Ex 38 Requiring Authentication if the caller isn't registered

Matching Patterns	Deploy Patterns
`$request = ^INVITE` `$registered( From ) = false`	`$auth = true`

Ex 39 Requiring Authentication if the call is from an IP address with the prefix "192.168.10"

Matching Patterns	Deploy Patterns
`$request = ^INVITE` `$addr = 192\.168\.\10`	`$auth = true`

Ex 40 Not Requiring Authentication from 10:00AM to 5:59PM

Matching Patterns	Deploy Patterns
`$request = ^INVITE` `$time =^1[0-7]:`	`$auth = false`

6. Load Balancing

You can change the routing destinations by using some unique value, for example, by time or by session ID. For example, you can reduce the load of a gateway by allocating the calls to multiple VoIP gateways.

Ex 41 Load Balancing by switching 3 destinations every second

Matching Patterns	Deploy Patterns
`$request = ^INVITE` `$time = [0369]$`	`$target = server1`
`$request = ^INVITE` `$time = [147]$`	`$target = server2`
`$request = ^INVITE`	`$target = server3`

Ex 42 Load Balancing by switching 2 destinations every 30 minutes

Matching Patterns	Deploy Patterns
`$request = ^INVITE` `$time = ^..:[0-2]`	`$target = server1`
`$request = ^INVITE`	`$target = server2`

Ex 43 Load Balancing based on whether the Session ID is odd or even

Matching Patterns	Deploy Patterns
`$request = ^INVITE` `$sid = [13579]$`	`$target = server1`
`$request = ^INVITE`	`$target = server2`

7. NAT Traversal

Brekeke SIP Server decides automatically whether to handle NAT traversal. The Administrator can also specify whether Brekeke SIP Server handles NAT traversal or not for each session using Dial Plan.

The following handling variables will be related to NAT traversal:

$nat = true or false

Whether to apply NAT traversal or not

$ifdst = IP address or FQDN

The interface address of Brekeke SIP Server for communicating with the session's destination (the callee).

This address is used for Via, Record-Route and etc.

$ifsrc = IP address or FQDN

The interface address of Brekeke SIP Server for communicating with the session's source (the caller).

This address is used for Via, Record-Route and etc.

7.1. Setting NAT Traversal ON/OFF

Ex 44 Enabling NAT Traversal if the callee's domain name is "host.domain"

Matching Patterns	Deploy Patterns
`$request = ^INVITE` `$geturi( To ) = @host.domain`	`$nat = true`

Ex 45 Disabling NAT Traversal if the call is from 192.168.0.1

Matching Patterns	Deploy Patterns
`$request = ^INVITE` `$addr = 192\.168\.0\.1$`	`$nat = false`

7.2. Specifying the Interface Address

Ex 46 **Using "192.168.1.1" as the interface address if the prefix of callee's contact address is "192.168.1"**

Matching Patterns	Deploy Patterns
`$request = ^INVITE` `$registeredaddr = ^192\.168\.1\.`	`$ifdst = 192.168.0.1.1`

Ex 47 **Using "192.168.2.1" as the interface address if the call is from an IP address with the prefix "192.168.2"**

Matching Patterns	Deploy Patterns
`$request = ^INVITE` `$ addr = ^192\.168\.2\.`	`$ifsrc = 192.168.0.2.1`

8. RTP Relay

Brekeke SIP Server decides whether to do RTP relay or not automatically for each session. The Administrator can also specify whether Brekeke SIP Server should do RTP relay or not using a Dial Plan. RTP relay will be enabled automatically for the session where NAT traversal is handled.

The following handling variables are related to RTP relay:

$rtp = true or false

Whether to apply RTP Relay or not

$ifdst = IP address or FQDN

The interface address of Brekeke SIP Server for communicating with the session's destination (the callee).
This address is used for receiving RTP packets from the destination UA.

$ifsrc = IP address or FQDN

The interface address of Brekeke SIP Server for communicating with the session's source (the caller).
This address is used for receiving RTP packets from the source UA.

Ex 48 Enabling RTP Relay if the callee's name prefix is "9"

Matching Patterns	Deploy Patterns
`$request = ^INVITE` `To = sip:9.+@`	`$rtp = true`

Ex 49 Enabling RTP Relay and using PCMA as the codec if the callee's SIP URI is sip:user@host

Matching Patterns	Deploy Patterns
`$request = ^INVITE` `$geturi(To) = sip:user@host`	`$rtp = true` `&net.rtp.audio.payloadtype = 8`

Ex 50 Enabling RTP Relay and assigning the range of ports from 10000 to 10100 if the is call from 192.168.0.1

Matching Patterns	Deploy Patterns

`$request = ^INVITE`
`$addr = 192\.168\.0\.1$`
	`$rtp = true`
	`&net.rtp.port.min = 10000`
	`&net.rtp.port.max = 10100`

9. Specifying Environment Variables

The environment variable is for setting the server's behavior, administrative information, various internal parameters.

To set an environment variable using the Dial Plan, please add the prefix '&' before the variable name. The environment variable's value set using the Dial Plan is valid only for the session the rule is applied.

Ex 51 Using G723 as the codec for all calls

Matching Patterns	Deploy Patterns
`$request = ^INVITE`	`&net.rtp.audio.payloadtype = 4`

Ex 52 Not Appending Record-Route header if the callee's name prefix is "9"

Matching Patterns	Deploy Patterns
`$request = ^INVITE` `To = sip:9.+@`	`&net.sip.addreordroute = false`

Ex 53 Not Adding lr parameter to Record-Route header if the callee's host name is "host"

Matching Patterns	Deploy Patterns
`$request = ^INVITE` `$geturi( To ) = @host`	`&net.sip.addreordroute.lr = false`

Ex 54 Not Appending rport parameter to Via header if the callee's host name is "host"

Matching Patterns	Deploy Patterns
`$request = ^INVITE` `$geturi( To ) = @host`	`&net.sip.addrport = false`

Ex 55 Setting the ringing timeout period to 30 seconds if the caller's name is "admin"

Matching Patterns	Deploy Patterns
`$request = ^INVITE` `From = sip:admin@`	`&net.sip.timeout.ringing = 30000`

Ex 56 Using Upper Registration to "host.domain" if the caller's name prefix is "9"

Matching Patterns	Deploy Patterns

`$request = ^REGISTER`
`From = sip:9.+@`
	`&net.registrar.upper.allow = true`
	`&net.registrar.upper.url = host.domain`
	`$continue = true`

Ex 57 Adjusting the following registration period as 100 seconds if the current period is less than 100 seconds

Matching Patterns	Deploy Patterns
`$request = ^REGISTER` `Expires = ^[0-9].$`	`&net.registrar.adjust.expires=100` `$continue = true`

Ex 58 Not Using Thru Registration if the callee's host name is "host"

Matching Patterns	Deploy Patterns
`$request = ^REGISTER` `$geturi( To ) = @host`	`&net.registrar.thru.allow = false` `$continue = true`

10. Using Session Plug-in

The Session Plug-in is a plug-in which is used for controlling sessions and collecting accounting information. The plug-in will be loaded by setting the plug-in name for the handling variable $session.

Ex 59 Using "RadiusAcct" plug-in for all calls

Matching Patterns	Deploy Patterns
`$request = ^INVITE`	`$session=com.sample.radius.proxy.RadiusAcct`

Ex 60 Using "CDRlog" plug-in if the callee's host name is "host"

Matching Patterns	Deploy Patterns
`$request = ^INVITE` `$geturi( To ) = @host`	`$session = com.user.CDRlog`