

Brekeke SIP Server

Version 2

Administrator's Guide

Brekeke Software, Inc.

Version

Brekeke SIP Server v2 Administrator's Guide

Revised July, 2009

Copyright

This document is copyrighted by Brekeke Software, Inc.

Copyright © 2002-2010 Brekeke Software, Inc.

This document may not be copied, reproduced, reprinted, translated, rewritten or readdressed in whole or part without expressed, written consent from Brekeke Software, Inc.

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 U.S 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	9
1.1. What is Brekeke SIP Server?.....	9
1.2. Editions	10
2. Installation	11
2.1. System Requirements.....	11
2.2. Installation for Windows with the Executable Installer	11
2.3. Installation for Linux, Mac OS X, and Solaris	13
3. Brekeke SIP Server Administration Tool	15
3.1. Server Status	15
3.1.1. Server Status	15
3.1.2. Database Status	16
3.2. Restart / Shutdown	16
3.3. Dial Plan	17
3.3.1. View Rules	17
3.3.2. New Rule/ Edit Rule	17
3.3.3. Import Rules	18
3.3.4. Export Rules	18
3.3.5. View Aliases.....	18
3.3.6. New Alias / Edit Alias	19
3.3.7. Import Alias.....	20
3.3.8. Export Alias	20
3.4. User Authentication	20
3.4.1. View Users	20
3.4.2. New User / Edit User	21
3.4.3. Import Users	21
3.4.4. Export Users.....	21
3.5. Registered Clients	22
3.5.1. View Clients	22
3.5.2. New Client.....	22
3.6. Active Sessions	23

3.6.1. Active Sessions.....	23
3.6.2. Session Details.....	24
3.7. Call Logs	26
3.7.1. Call Logs	26
3.7.2. Daily Log	27
3.8. Configuration	27
3.8.1. System.....	27
1) General	28
2) Network	28
3) Address Filtering.....	29
4) UPnP.....	29
5) Java	29
3.8.2. SIP.....	30
1) SIP exchanger.....	30
2) NAT traversal	30
3) Authentication	30
4) Upper Registration	31
5) Thru Registration	32
6) Timeout	32
7) Miscellaneous.....	32
8) TCP	32
9) Performance Optimization (Proxy).....	33
10) Performance Optimization (Registrar)	33
3.8.3. RTP	33
1) RTP exchanger	33
2) Timeout	34
3.8.4. Database/Radius	34
1) Embedded Database.....	35
2) Thirdparty Registered Database	35
3) Thirdparty Users Database.....	35
4) Thirdparty Alias Database	36
5) Radius	36
3.8.5. Password	36
3.8.6. Domains	36
3.8.7. New Domain / Edit Domain.....	38

3.8.8. Mirroring.....	38
1) Server Status	39
2) Mirroring Settings	39
3.8.9. Heartbeat.....	40
1) Heartbeat Status	40
2) Heartbeat Settings	41
3) Remote Access.....	41
3.8.10.Heartbeat Settings	42
3.8.11.Action Settings	42
1) Send Email	43
2) Re-initialize as primary	43
3) Add IP Address.....	44
4) Delete IP Address	44
5) Execute Command	45
6) Management Command.....	45
3.8.12.Advanced	45
3.9. Maintenance.....	46
3.9.1. Back Up.....	46
3.9.2. Restore.....	46
3.9.3. Update Software	46
3.9.4. Activate	46
4. Dial Plan.....	47
4.1. What is the Dial Plan?	47
4.2. Create and Edit Dial Plan	47
4.3. Matching Patterns	48
4.3.1. Syntax.....	48
1) SIP Header Field Name	49
2) Environment Variable	49
3) Conditional Function	50
4.3.2. Reference of Conditional Functions	51
1) General Functions.....	51
\$addr	51
\$body.....	52

<code>\$date</code>	52
<code>\$geturi</code>	53
<code>\$globaladdr</code>	53
<code>\$headerparam</code>	54
<code>\$istalking</code>	54
<code>\$mirroring</code>	55
<code>\$mydomain</code>	56
<code>\$not</code>	56
<code>\$outbound</code>	56
<code>\$param</code>	57
<code>\$port</code>	58
<code>\$primary</code>	58
<code>\$registered</code>	58
<code>\$registeredaddr</code>	59
<code>\$registereduri</code>	59
<code>\$regaddr</code>	59
<code>\$reguri</code>	60
<code>\$request</code>	60
<code>\$sid</code>	61
<code>\$sessionnum</code>	61
<code>\$soapget</code>	61
<code>\$subparam</code>	62
<code>\$time</code>	62
<code>\$transport</code>	63
<code>\$uriparam</code>	63
<code>\$webget</code>	64
2) Alias Functions	64
<code>\$alias.lookup</code>	64
<code>\$alias.reverse</code>	65
3) Mathematical Functions	65
<code>\$math.ge</code>	65
<code>\$math.gt</code>	66
<code>\$math.le</code>	66
<code>\$math.lt</code>	66
<code>\$math.rand</code>	67

4) String Functions	67
\$str.equals	67
\$str.hashcode	67
\$str.isdigits	68
\$str.length	68
\$str.md5	68
\$str.remove	69
\$str.reverse	69
\$str.substring	69
\$str.trim	70
5) User Directory Functions	70
\$usrdir.lookup	70
4.4. Deploy Patterns	71
4.4.1. Syntax.....	71
1) SIP Header Field Name	71
2) Environment Variable	72
3) Handling Variable	73
4.4.2. Reference of Handling Variable	73
\$action.....	73
\$auth.....	74
\$b2bua.....	74
\$continue	74
\$ifdst.....	75
\$ifsrc.....	75
\$log.....	76
\$nat.....	76
\$replaceuri.from.....	76
\$replaceuri.to	77
\$request	77
\$response	77
\$rtp	78
\$session.....	78
\$target	79
5. Upper Registration and Thru Registration.....	80

5.1. Upper Registration	80
5.2. Thru Registration.....	81
6. NAT Traversal	82
6.1. Brekeke SIP Server Behind NAT (Near-End NAT traversal).....	82
6.1.1. UPnP Settings.....	82
6.1.2. Manual Configuration	82
6.2. For Clients Behind NAT over the Internet (Far-End NAT traversal)	83
7. Basic Setup.....	85
7.1. Setup Brekeke SIP Server.....	85
7.2. Setup SIP Client.....	85
7.3. Make a test call	86
8. Security	87
8.1. Administration Tool.....	87
8.2. SIP Authentication.....	87
8.2.1. SIP Authentication for all INVITE/REGISTER requests	87
8.2.2. SIP Authentication for certain requests.....	87
8.3. To block a non-registered user's INVITE request.....	88
9. Mirroring/Heartbeat.....	89
9.1. Deployment Structure	89
9.2. The Primary Server Settings	90
9.2.1. Firewall Settings at the Primary Server	90
9.2.2. Add the Virtual IP Address in the Primary Server.....	90
9.2.3. Mirroring Settings at the Primary Server	90
9.3. The Secondary Server Settings	90
9.3.1. Mirroring Settings at the Secondary Server.....	90
9.3.2. Heartbeat Settings for the Secondary Server	91
9.4. Start the Mirroring and Heartbeat features	92
9.4.1. Start the Primary Server	92

9.4.2. Start the Secondary Server	92
10. Environment Variables	93
10.1. General	93
10.2. Registrar	93
10.3. TCP	94
10.4. UPnP	94
10.5. Logging	95
11. Uninstall	96
11.1. Uninstall from Windows.....	96
11.2. Uninstall from Linux, Mac OS X, and Solaris.....	96
Appendix A: Glossary	97

1. Introduction

This document explains the installation and configuration settings of the Brekeke SIP Server Software. The document will help you to start a SIP based service such as VoIP (Voice over IP).

1.1. What is Brekeke SIP Server?

The Brekeke SIP Server is an open standard based SIP Proxy Server and Registrar. It authenticates and registers user agents such as VoIP device and softphone, and routes SIP sessions such as VoIP calls between user agents.

The Brekeke SIP Server has the following main functions:

◆ Routing

The Brekeke SIP Server will route SIP requests from a SIP user agent or another server to the most appropriate SIP URI address based on its Registrar Database. By specifying desired routing settings in the Dial Plan, you can also prioritize your routing. If the routing resolves successfully on the server, you can establish a session even when the final SIP URI address is unknown to the caller. Using Regular Expressions, you can easily create a Dial Plan rule that will analyze SIP headers or the IP address of SIP packets to route calls. For example, you can set a prefix for each location with Dial Plan settings. Such settings are especially useful for multi-location office usage of the Brekeke SIP Server.

◆ Registrar

The Brekeke SIP Server receives REGISTER requests from SIP user agents, and updates its Registrar Database. SIP URI in the REGISTER request will be added in the database as a user's address. Using the registrar function, you will be able to receive calls from any SIP user agents using your unique SIP URI.

◆ NAT Traversal

When caller and callee are located on different networks, the Brekeke SIP Server can connect calls by rewriting SIP packets appropriately. It is common to have private local IP addresses within a LAN environment, thus NAT Traversal service is necessary when a local user is establishing a connection with another user in the global IP network (Internet). Depending upon the situation, Brekeke SIP Server will relay RTP packets to prevent losing media data such as voice and video. The NAT traversal feature on the Brekeke SIP Server

supports both Near-End NAT (the server and SIP user agents located within the same firewall) and Far-End NAT (SIP user agents located on the other side of a firewall of a remote network).

◆ Upper/Thru Registration

Upper/Thru Registration is a unique feature of the Brekeke SIP Server that allows easy configuration of parallel users of pre-existing or other SIP servers. By forwarding REGISTER requests to specified SIP servers, the feature allows users to register their SIP user agents at the other SIP server and the Brekeke SIP Server simultaneously. For example, with this feature, users can register their SIP user agents at an ITSP, thus users under the Brekeke SIP Server can talk with other users in the ITSP or receive calls from PSTN.

1.2. Editions

The Brekeke SIP Server comes in several editions to meet the needs of different levels of users.

Edition	Explanation	Common type of usage
Advanced	It is to be used by commercial users and by general users.	Carrier Class, Service Providers
Standard	It is to be used by commercial users and by general users.	Business phone system, General Commercial Use, Training, R&D, etc
Evaluation	It may be used by anyone who wishes to internally evaluate the product during the Evaluation Period. This license is free of charge.	Product Trial prior to purchase
Academic	It may be used only by students and faculty members or staff members of a degree-granting educational institution (elementary schools, middle or junior high schools, high schools, junior colleges, colleges, and universities). This license is free of charge.	Training, student projects, technology lab

2. Installation

Brekeke SIP Server can work on Microsoft Windows, Linux, Mac OS X and Solaris. There are two ways to install the product. For Windows, an administrator can use the executable installer. For all other platforms, an administrator needs to copy an installation package file into Tomcat.

2.1. System Requirements

The Brekeke SIP Server supports the following platforms:

OS	Microsoft Windows, Linux, Mac OS X, Solaris
Java	Version 1.5 or later <i>Note: We recommend using Java provided by Sun Microsystems.</i>
Apache Tomcat	Version 5.5.12 or later <i>Note: Tomcat is not required if the installer for Windows is used.</i>
Memory	At least 256 MB

2.2. Installation for Windows with the Executable Installer

Step 1: Install Java

Download Java from the following Sun Microsystems' website and install it.

<http://www.java.com/>

If you already have Java in your computer, please make sure that the Java' version is 1.5 or later.

We recommend using Java provided by Sun Microsystems.

Step 2: Install the Brekeke SIP Server

1. Obtain the executable installer and a Product ID for Brekeke SIP Server.
2. Start the installer.
3. Continue the installation by following the installer's instructions. The Brekeke SIP Server will be installed automatically. If you check the **[Run Brekeke SIP Server]** box at the last stage of the installation and push the **[Finish]** button, the Brekeke SIP Server's HTTP service will start automatically.

Step 3: Start the Brekeke SIP Server's HTTP service

If you did not check **[Run Brekeke SIP Server]** at the last stage of the installation, please start

the Brekeke SIP Server's HTTP Service by the following steps.

1. Open **[Control Panel] > [Administrative Tools] > [Service]**.
2. Select **[Brekeke SIP Server]** and start the service.
3. Restart your computer.

The Brekeke SIP Server's HTTP service will automatically start.

Step 4: Start the Brekeke SIP Server Administration Tool

1. Select **[Start] > [All Programs] > [Brekeke] > [Brekeke SIP Server] > [Brekeke SIP Server Admintool]**.

A web browser will open and you will see the License Agreement page. Enter the Product ID you have in the **[Product ID]** field. Push **[I agree]** button and then on **[Activate]** button to activate the License.

Note: You will need to activate the Product ID only when you are freshly installing version 2.x or upgrading the product from OnDO SIP Server version 1.5 to Brekeke SIP Server version 2.x. For all other Brekeke SIP Server upgrade do not need Product ID to be activated.

2. At the Admintool Login page, enter User ID and Password and push **[Login]** button. The default administrator's User ID is "sa" and its Password is "sa".
3. After the login, push the **[Start]** button. If the Status is **[Active]**, the Brekeke SIP Server has started successfully.
If the Status is **[Inactive]**, the server has not started successfully. The error should be shown above the Status.

*Note: When the Brekeke SIP Server's port number (default port 5060) is already in use by another application, the server status will be shown as **[Inactive]**. For example, if you attempt to start the server while another SIP UA is running on the same computer, the server may fail to start. In this case, please stop the other SIP UA, and click the **[start]** button on the Admintool's **[Restart/Shutdown]** page.*

2.3. Installation for Linux, Mac OS X, and Solaris

Step 1: Install Java

Download Java from the following website and install it.

For Windows, Linux and Solaris: <http://java.sun.com/javase/downloads/>

For Mac OS X: <http://developer.apple.com/java/>

If you already have Java in your computer, please make sure that the Java's version is 1.5 or later.

Step 2: Install Apache Tomcat

Download Tomcat from the following website and install it. <http://jakarta.apache.org/tomcat/>

If you already have Tomcat in your computer, please make sure that the Tomcat's version is 5.5.12 or later.

- ***We recommend adding `liveDeploy="false"` to the `server.xml` file at Tomcat installation directory/conf/ as shown below.***

```
<Host name="localhost" appBase="webapps"
      unpackWARs="true" autoDeploy="true" liveDeploy="false"
      xmlValidation="false" xmlNamespaceAware="false">
```

Step 3: Install the Brekeke SIP Server

1. Obtain the installation package file (.war file) and a Product ID.
2. Copy the war file into the directory "webapps", which is located under the Tomcat installation directory.

Step 4: Start the Brekeke SIP Server Administration Tool

1. Start the Tomcat.
2. Open a web browser and access the URL `http://localhost:18080/proxy`
(If you chose an http port number other than "18080" when installing the Tomcat, change the port number in the URL above to the number specified during installation).
3. You will see the License Agreement page. Enter the Product ID you have in the **[Product ID]** field. Push **[I agree]** button and then on **[Activate]** button to activate the License.

Note: You will need to activate the Product ID only when you are freshly installing version 2.x or upgrading the product from OnDO SIP Server version 1.5 to Brekeke SIP Server

version 2.x. For all other Brekeke SIP Server upgrade do not need Product ID to be activated.

4. At the Admintool Login page, enter User ID and Password and push **[Login]** button. The default administrator's User ID is "sa" and its Password is "sa".
5. After the login, push the **[Start]** button. If the Status is **[Active]**, the Brekeke SIP Server has started successfully.
6. If the Status is **[Inactive]**, the server has not started successfully. The error should be shown above the Status.

*Note: When the Brekeke SIP Server's port number (default port 5060) is already in use by another application, the server status will be shown as **[Inactive]**. For example, if you attempt to start the server while another SIP UA is running on the same computer, the server may fail to start. In this case, please stop the other SIP UA, and click the **[start]** button on the Admintool's **[Restart/Shutdown]** page.*

3. Brekeke SIP Server Administration Tool

Brekeke SIP Server Administration Tool (Admintool) is a web based GUI application which allows administrators to manage Brekeke SIP Server. This section provides reference information for the tool.

To login to the Administration Tool, the correct User ID and Password are required. The default administrator's User ID is "sa" and its Password is "sa".

3.1. Server Status

The Server Status page shows the version information and current status of the server and databases. Some of these values can be modified through the [Configuration] menu.

3.1.1. Server Status

Field Name	Explanation
Status	If the SIP Server is running, the status is "ACTIVE". Otherwise, the status is "INACTIVE".
server-product	Product name
server-ver	Version and revision number
server-name	Server name
server-description	Description
server-location	Location
server-startup-time	Time the server was started
server-current-time	Server's current time
server-life-length	Length of time the server has been running for
machine-name	Hostname
listen-port	SIP listen port
transport	Acceptable transport type
interface	Network interface IP address(es) used by the server
startup-user	User name that started the server
work-directory	The directory path that server is running from
session-active	The number of currently active sessions
session-total	The total number of sessions processed
session-peak	The number of peak sessions

Field Name	Explanation
registered-record	Number of records in Registrar Database
os-name	OS name
os-ver	OS version
java-ver	Java version
admin-sip	Administrator's SIP URI
admin-mail	Administrator's e-mail address

3.1.2. Database Status

Field Name	Explanation
registered-database	Status of the connection with Registrar Database
userdir-database	Status of the connection with User Directory Database
alias-database	Status of the connection with Alias Database

3.2. Restart / Shutdown

An administrator is able to start or shutdown the Brekeke SIP Server in the Restart/Shutdown page.

Field Name	Explanation
Status	If the server is running, the status is "ACTIVE". Otherwise, the status is "INACTIVE".
Listen-Port	SIP listen port
Active Sessions	The number of currently active sessions

Button	Explanation
Start	Starts Brekeke SIP Server
Restart	Restarts Brekeke SIP Server
Shutdown	Stops Brekeke SIP Server. A message to confirm the shutdown command will appear if there are any active sessions. Selecting [Force Shutdown] will terminate all active sessions and shutdown the server.




3.3. Dial Plan

The Dial Plan menu is for editing Dial Plan rules and Alias Database. Please refer to Section 4 for details about the Dial Plan syntax.

3.3.1. View Rules

The View Rules page shows the list of existing Dial Plan rules. The rule in the higher position in the list has the higher priority. Disabled rules are shown in grey. The buttons at the right side of each rule are for editing, copying and deleting the rule. By pressing the [Apply Rules] button, you can apply the new rules or modified rules even while the server is running.

Field Name	Explanation
Priority	Priority of the Dial Plan rule
Name	The name of the Dial Plan rule
Matching Patterns	Defined condition
Deploy Patterns	How the SIP request should be processed





Button	Explanation
 Edit	Edit the Dial Plan rule
 Copy	Copy the Dial Plan rule
 Delete	Delete the Dial Plan rule
Apply Rules	Save and apply changes

3.3.2. New Rule/ Edit Rule

New Rule page helps an administrator to create a new Dial Plan rule. Edit Rule page helps an administrator to modify an existing Dial Plan rule.

Item	Explanation
Rule name	Name of the rule
Description	Description of the rule
Priority	Priority of the rule
Disabled	When it is checked, the rule is disabled.

Item	Explanation
Matching Patterns	List of Matching Patterns Please refer to the “4.3. Matching Patterns”.
Deploy Patterns	List of Deploy Patterns Please refer to the “4.4. Deploy Patterns”.
Variable	The name of variable By pressing [...] button, most of the variables are displayed for you to choose from.
Value	For Matching Patterns: a value of the variable that should match For Deploy Patterns: the value that will be assigned to the variable

Button	Explanation
 Insert	Insert the specified definition in [Variable] and [Value] fields into the given list box.
 Delete	Delete the selected definition. The deleted definition is displayed in [Variable] and [Value] fields.
 Down	Move the selected definition down
 Up	Move the selected definition up
Save	Save the Dial Plan rule and return to the [View Rules] page
Cancel	Cancel the changes and return to [View Rules] page

3.3.3. Import Rules

You can import and upload new Dial Plan rules with the Import Rules option.

Select a Dial Plan table file to import Dial Plan rules from [Browse...] button and then click the [Upload] button to upload Dial Plan rules.

3.3.4. Export Rules

You can export the existing Dial Plan rules to another location using the Export Rules option.



3.3.5. View Aliases

The View Aliases page shows the list of alias records stored in the Alias Database. The buttons at the right side of each record are for editing, copying and deleting the record.

To lookup the record from the Alias Database, please use \$alias.lookup or \$alias.reverse conditional function in Matching Patterns.

Note: The Alias feature is available in the Advanced Edition only.

Field Name	Explanation
Alias Name	Alias name of the record
Group ID	Optional ID for a group of Alias records
Entity Name	Entity name of the record

Button	Explanation
 Edit	Edit the record
 Delete	Delete the record

Filter	
Item	Explanation
Pattern	Search keywords
On Field	By: Alias Name, Group ID, Entity Name
Maximum Rows	Number of results to display

3.3.6. New Alias / Edit Alias

New Alias page helps an administrator to create a new alias record. Edit Alias page helps an administrator to modify an existing alias record.

Note: Alias feature is available in the Advanced Edition only.

	Item	Explanation
*	Alias	Alias name of the record
	Group ID	Optional ID for a group of Alias records
*	Entity	Entity name of the record

(is a required field.)*

Button	Explanation
Modify	Modify the alias record
Add	Add the alias record
Delete	Delete the alias record

3.3.7. Import Alias

You can import and upload new alias records with the Import Alias option.

Select an alias record file in the CSV format from [Browse...] button and then click the [Upload] button to upload alias records.

The CSV format:

```
Alias_Name, [Group_ID], Entity_Name
```

3.3.8. Export Alias

You can export the existing alias records to another location using the Export Alias option. The records will be saved in the CSV format.

3.4. User Authentication

The User Authentication is for adding and editing a user for authentication. The setting for enabling authentication is at the Configuration page. Refer to the section “3.8.2 SIP” for more details.

3.4.1. View Users

The View Users page shows the list of existing users for authentication. The buttons at the right side of each user are for editing and deleting the user.

Field Name	Explanation
User	User name for authentication
Name	User's long name
Email Address	User's e-mail address
Description	Misc. User information

Filter	
Item	Explanation
Pattern	Search keywords
On Field	By: User, Name, Email Address, Description
Maximum Rows	Number of results to display

3.4.2. New User / Edit User

New User page helps an administrator to create a new user for authentication. Edit User page helps an administrator to modify an existing user.

	Item	Explanation
*	User	Username for authentication
*	Password	Password
*	Confirm Password	Reenter Password
	Name	User's name
	Email Address	User's e-mail address
	Description	Misc. User information

(* is a required field.)

Button	Explanation
Modify	Modify the user record
Add	Add the user record
Delete	Delete the specified user record

3.4.3. Import Users

You can import and upload new user information with the Import Users option.

Select a user record file in the CSV format from [Browse...] button and then click the [Upload] button to upload user records.

The CSV format:

```
User, [Password], [Name], [Email Address], [Description]
```

3.4.4. Export Users

You can export the existing user information to another location using the Export Users option.

The records will be saved in the CSV format.

3.5. Registered Clients

The Registered Clients page is for viewing and adding registered SIP clients to the Registrar Database.

3.5.1. View Clients

This page displays the registered SIP client records that are in the Registrar Database. When the Brekeke SIP Server accepts a REGISTER request from a SIP client, the database is updated automatically. The button at the right side of each record is for deleting the record.

Field Name	Explanation	
User	Username	
Contact URI	User's contact SIP URI	
Detail	Details of the registration record	
	Variable	Explanation
	Expires	Expiration period of the record [seconds]
	Priority	Priority of the record (100 - 1000)
	User Agent	Name of the client's product if available.
	Requester	The IP address that this REGISTER request was sent from.
	Time Update	Timestamp of the latest update of the record

Filter	
Item	Explanation
Pattern	Search keywords
On Field	By: User, Contact URI, Requester

3.5.2. New Client

The New Client page allows an administrator to create a registration record manually.

	Item	Explanation
*	User	Use name that receives a contact from other UAs.
*	Contact URI	User's contact SIP URI. It should contain a reachable IP address.
*	Expires	The length in seconds that a record will be stored in the Registrar Database. Records will be deleted after the specified time passes. While the record is stored in the database, registered users can receive contacts from other SIP UAs through the specified username that was designated in the "User" setting.
*	Priority	Priority of the record (100 - 1000).

(* is a required field.)

Button	Explanation
Register	Register a new record in the registrar database.

3.6. Active Sessions

The Active Sessions shows currently active SIP sessions and their details. It also allows an administrator to end a certain session.

3.6.1. Active Sessions

The Active Sessions page shows the list of currently active SIP sessions. The buttons at the right side of each session is for viewing the details of the session.

Field Name	Explanation
Session ID	Session ID
From	UAC's SIP URI and its IP address
To	UAS's SIP URI and its IP address
Time	Session start time

Field Name	Explanation		
Status	Session Status		
	Status	Explanation	Trigger
	Initializing	Initializing a new session	
	Inviting	Sending an initial request	An initial request
	Provisioning	Preparing for setting up a session	1xx response
	Ringling	Ringling	18x response
	Accepted	Established	2xx response
	Talking	Talking	ACK request
	Closing	Closing	BYE or error response

Filter	
Item	Explanation
From	UAC's SIP URI or its IP address
To	UAS's SIP URI or its IP address
Time Range	Time period
Status	Session status

3.6.2. Session Details

The session details page displays detailed information for the selected SIP session.

Field Name	Explanation
EX-SID	Internal Session Thread ID
From-uri	UAC's SIP URI
From-ip	UAC's IP address and the transport
From-if	Network interface address of UAC's side

Field Name	Explanation																								
To-uri	UAS's SIP URI																								
To-ip	UAS's IP address and the transport																								
To-if	Network interface address of UAS's side																								
Call-ID	Call-ID																								
rule	Dial Plan rules which are applied for the session																								
plug-in	Session Plug-in used to handle the session																								
sip-packet-total	Total number of received SIP packets																								
listen-port	SIP listen port																								
session-status	<table border="1"> <thead> <tr> <th>Status</th> <th>Explanation</th> <th>Trigger</th> </tr> </thead> <tbody> <tr> <td>Initializing</td> <td>Initializing a new session</td> <td></td> </tr> <tr> <td>Inviting</td> <td>Sending an initial request</td> <td>An initial request</td> </tr> <tr> <td>Provisioning</td> <td>Preparing for setting up a session</td> <td>1xx response</td> </tr> <tr> <td>Ringing</td> <td>Ringing</td> <td>18x response</td> </tr> <tr> <td>Accepted</td> <td>Established</td> <td>2xx response</td> </tr> <tr> <td>Talking</td> <td>Talking</td> <td>ACK request</td> </tr> <tr> <td>Closing</td> <td>Closing</td> <td>BYE or error response</td> </tr> </tbody> </table>	Status	Explanation	Trigger	Initializing	Initializing a new session		Inviting	Sending an initial request	An initial request	Provisioning	Preparing for setting up a session	1xx response	Ringing	Ringing	18x response	Accepted	Established	2xx response	Talking	Talking	ACK request	Closing	Closing	BYE or error response
	Status	Explanation	Trigger																						
	Initializing	Initializing a new session																							
	Inviting	Sending an initial request	An initial request																						
	Provisioning	Preparing for setting up a session	1xx response																						
	Ringing	Ringing	18x response																						
	Accepted	Established	2xx response																						
	Talking	Talking	ACK request																						
Closing	Closing	BYE or error response																							
time-inviting	Session start time																								
time-talking	Talking start time																								
length-talking	Length of talking																								
rtp-relay	RTP relay status																								

When RTP relay is enabled, the [rtp-relay] field shows “on” and the information below is displayed. This information shows status of RTP streams of both [rtp-srcdst] (UAC to UAS) and [rtp-dstsrc] (UAS to UAC).

Field Name	Explanation
media	Media type (audio, video)
transport	Transport type
payload	Payload type
status	Status (active, hold)
listen-port	UDP port number for receiving RTP packets
send-port	UDP port number for sending RTP packets

Field Name	Explanation
target	Destination IP address of RTP packets
packet-count	The number of packets
packet/sec	The number of packets per seconds
current size	Packet size (bytes) of RTP sent most recently
buffer size	Buffer size (bytes)

Button	Explanation
Disconnect	Disconnects the SIP session
Back	Go back to the [Active Sessions] page

3.7. Call Logs

The Call Logs is for showing the call logs.

3.7.1. Call Logs

The Call Logs page shows the calendar with the number of sessions by date. Please click the desired date to display that date's session log.

Check Box	Explanation
HTML	Clicking a date will display that daily log page in a new browser window.
CSV	Clicking a date will save that daily log in a CSV file.

Button	Explanation
Save	Specify a term to save logs for. Logs older than the specified term will be deleted automatically.

The CSV format:

```
SID,FromURI,ToURI,TalkingLength,InvitingStart,TalkingStart,SessionEnd,Result,ErrorCode
```

3.7.2. Daily Log

A daily session log will be displayed in a new window. You can filter the call logs by stating the From-URI to To-URI.

Field Name	Explanation
sid	Session ID
from-url	UAC's SIP URI
to-url	UAS's SIP URI
talking-length	Talking time
invite-start-time	Session start time
talk-start-time	Talking start time
end-time	Session end time
result	Result
error	Error Code "-1" indicates a normally ended call. For irregularly ended calls, a SIP error response code will be displayed.

Filter	
Item	Explanation
From-Url	UAC's SIP URI
To-Url	UAS's SIP URI
Time Range	Time period
Maximum Rows	Number of results to display

3.8. Configuration

The Configuration is for editing settings, managing database and domains, and updating the software. Changes will take effect when the SIP server is restarted.

The Mirroring and Heartbeat features provide High Availability (HA) functions and keep your SIP service alive.

3.8.1. System

The System page allows an administrator to configure a system and general network settings.

1) General

Item	Default value	Explanation
Server Name	your-sip-sv	Name of the server
Server Description	your SIP Server	Description for the server
Server Location	your-place	Location of the server
Administrator SIP URI	your-sip-uri	Administrator's SIP URI
Administrator Email Address		Administrator's e-mail address
Start up	auto	When "auto" is set, Brekeke SIP Server will automatically start when the web server (Tomcat) is started.

2) Network

Item	Default value	Explanation
Interface address 1-5		<p>IP address(es) or FQDNs to be used as interface address(es) by Brekeke SIP Server. They will be shown in "interface" field of the [Server Status] page.</p> <p>IP addresses which can be used as interface addresses are the IP addresses assigned to the Network Interface Cards (NIC) of the computer where Brekeke SIP Server is installed.</p> <p><i>Note: In a Windows, Mac OS X and certain environments, Brekeke SIP Server will automatically get the local IP address.</i></p> <p>When the server is located behind a NAT, an administrator may need to specify the global IP address or its FQDN of the NAT here.</p> <p><i>Note: If the UPnP is enabled, Brekeke SIP Server will automatically find a router and get the global IP address.</i></p>
DNS caching period	3600	Period for which result of DNS name resolution will be kept. When "-1" is set, the record will be kept forever and the cache will not be refreshed. [seconds]

Item	Default value	Explanation
Auto interface discovery	off	When it is set for "on", interface address will be updated automatically.
External IP address pattern		Regular-expression based IP address pattern which should be treated as an external IP address.

3) Address Filtering

Item	Default value	Explanation
IP address filter	disable	When it is set for "allow", SIP Server will accept SIP packets only from the IP address specified in the Filter Pattern field. When it is set for "block", SIP Server will accept SIP packets from IP addresses other than the IP address specified in the Filter Pattern field.
Filter pattern		Specify the desired IP address pattern by Regular Expressions.

4) UPnP

For using the UPnP feature, please use a router which supports UPnP and enable it at the settings of the router.

Item	Default value	Explanation
Enable/Disable	disable	When it is set for "enable", SIP Server will use UPnP to discover a router, to recognize the global IP address, and to manage port-forwarding.
Default router IP address		The local IP address of the router
Cache size	24	Size of the internal port-mapping cache table.
Cache period	86400	Cache period of the port-mapping. When "0" is set, the caching will be disabled. [seconds]
Refresh Interval	30	Refresh interval period of the UPnP. When "0" is set, the refresh will be disabled. [seconds]

5) Java

Item	Default value	Explanation
Java VM arguments		Specify parameters (excluding "classpath") that will be passed to the Java VM.

3.8.2. SIP

Configure SIP settings, NAT traversal, Authentications, Performance Optimizations and various timeout settings.

1) SIP exchanger

Item	Default value	Explanation
Session Limit	-1	Maximum number of SIP sessions the server will handle concurrently. "-1" specifies an unlimited number of SIP sessions.
Local Port	5060	Port number to send/receive SIP packets. Please use the default value of 5060 if you don't have any specific reason for changing this port.
B2B-UA mode	off	When set to "on", the B2B-UA mode will be enabled. With the B2B-UA mode, Brekeke SIP Server hides Via and Record-Route headers and replaces the original Call-ID header with a unique value.

2) NAT traversal

For the details of NAT traversal, please refer to the section "6. NAT Traversal".

Item	Default value	Explanation
Keep address/port mapping	on	When set to "on", the Brekeke SIP Server will send keep-alive packets to SIP UAs that are behind NAT at specified intervals. This is so that NAT will not close the external port used by the server to send packets to SIP UAs that are behind NAT.
Interval	12000	Interval for above setting. [milliseconds] If the server can not reach a SIP UA that is behind NAT, please set a shorter value here.
Add'rport' (Send)	on	When "on" is set, the server adds "rport" in Via header of an outgoing request packet so that the server can detect its own port number.
Add'rport' (Receive)	off	When "on" is set, the server adds "rport" with the value of the sender's source port in Via header of an incoming request packet.

3) Authentication

After REGISTER or INVITE authentication is enabled, an administrator needs to add users in the **[User Authentication]** page. Refer to the section "3.4. User Authentication" for more details.

Item	Default value	Explanation
REGISTER	on	When set to "on", the Brekeke SIP Server authenticates REGISTER requests.
INVITE	on	When set to "on", the Brekeke SIP Server authenticates INVITE requests.
Realm		This is set as the "realm" value. If left blank, the server's IP address is used as the "realm".
Auth-user=user in "To:" (Register)	yes	When set to "yes", the Brekeke SIP Server will authenticate REGISTER requests only when authentication user name matches the user name in the To header. When set to "no", the Brekeke SIP Server will authenticate all REGISTER requests.
Auth-user=user in "From:"	yes	When set to "yes", the Brekeke SIP Server will authenticate requests only when authentication user name matches the user name in the From header. When set to "no", the Brekeke SIP Server will authenticate all requests.
FQDN only	no	When set to "yes", only SIP URIs that contain an FQDN will be accepted. SIP URIs that contains IP addresses will not be accepted.
Nonce Expires	60	The length of the nonce expiration for authentication. [[seconds]

4) Upper Registration

See the section "Upper Registration" for more details.

Item	Default value	Explanation
On/Off	off	Enable/disable Upper Registration
Register Server		IP address or FQDN of a register server to be used as the Upper Registration destination
Protocol	UDP	Transport protocol used for upper registration UDP or TCP

5) Thru Registration

Item	Default value	Explanation
On/Off	on	Enable/disable Thru Registration

6) Timeout

Item	Default value	Explanation
Ringing Timeout	240000	Timeout for ringing time [milliseconds]
Talking Timeout	259200000	Timeout for talking time [milliseconds]
Upper/Thru Timeout	30000	Timeout for waiting the response for a REGISTER request to Upper Registration/Thru Registration destination [milliseconds]

7) Miscellaneous

Item	Default value	Explanation
100 Trying	any requests	When “any requests” is selected, the SIP Server will send back 100 Trying response for any initial request. When “only for initial INVITE” is selected, the SIP Server will send back 100 Trying response for initial INVITE request only.
Server/User-Agent		The specified name will be shown in Server and User-Agent headers. <i>Note: This feature is available in the Advanced Edition only.</i>

8) TCP

TCP feature is not available in the Academic Edition. Please refer the section “8.3. TCP” for specific configuration.

Item	Default value	Explanation
TCP-handling	on	Enable/disable TCP-handling
Queue Size	50	The size of the TCP connection queue
UDP Failover	on	When set to "on", the SIP Server uses an UDP connection after the TCP connection fails

9) Performance Optimization (Proxy)

Item	Default value	Explanation
Initial threads	10	Maximum number of pre-created (pooled) threads for the proxy service.
Maximum Sessions per thread	10	Maximum number of sessions per thread for the proxy service.

10) Performance Optimization (Registrar)

Item	Default value	Explanation
Initial threads	10	Maximum number of pre-created (pooled) threads for the registrar service.
Maximum Sessions per thread	10	Maximum number of sessions per thread for the registrar service.

3.8.3. RTP

The RTP page allows an administrator to configure RTP settings. If NATs are involved in the SIP communications, Brekeke SIP Server will relay RTP packets so that the RTP packets reach the SIP clients which are behind NAT.

1) RTP exchanger

Item	Default value	Explanation
RTP relay	auto	When set to "on", RTP packets will be relayed through the Brekeke SIP Server. When set to "auto", Brekeke SIP Server will decide whether or not to relay RTP automatically. For example, when Brekeke SIP Server detects a NAT, RTP packets are automatically relayed.
RTP relay (UA on this machine)	auto	When set to "auto", the server will decide automatically whether to relay RTP or not. When set to "off", Brekeke SIP Server will not relay RTP packets for the clients running on the server computer.
Minimum Port	10000	The minimum UDP port number to transmit RTP packets from.
Maximum Port	10999	The maximum UDP port number to transmit RTP packets from.

Item	Default value	Explanation
Minimum Port (Video)	0	The minimum UDP port number to transmit RTP packets for Video stream from. If set to "0", the server uses the same port range as Audio.
Maximum Port (Video)	0	The maximum UDP port number to transmit RTP packets for Video stream from. If set to "0", the server uses the same port range as Audio.
Port mapping	source port	Selects a destination port number for the Brekeke SIP Server to send RTP packets to clients behind Far-End NAT. Designates whether to use the source port from RTP packet (when set to "Source Port") or the RTP port specified in SDP (when set to "sdp").

2) Timeout

Item	Default value	Explanation
RTP Session Timeout	600000	The timeout for detecting RTP packets when relaying RTP. [milliseconds]

3.8.4. Database/Radius

The Database/Radius page allows an administrator to configure database and Radius settings. Here is the list of the databases which Brekeke SIP Server uses.

Database Name	Purpose
Registered Database	Registered Table This table stores the data of registered user agents. The data will be updated by REGISTER requests and used for the session routing. The [Registered Clients] page shows the list of registered user agents. Please refer to the section "3.5. Registered Clients".
Users Database	Users Table This table stores authentication data of users. The [User Authentication] page shows the list of users. Please refer to the section "3.4. User Authentication".
Alias Database	Alias Table This table stores alias data. The [View Aliases] page shows the list of alias. Please refer to the section "3.3.5. View Aliases". <i>Note: Alias Database is available in Advanced Edition only.</i>

Each database can use Embedded or Third-Party database system. Please refer to “Using a Third-Party Database Tutorial” for more information about using of Third-Party database system.

1) Embedded Database

Item	Default value	Explanation
Port Number	9001	TCP port number used by the Embedded database system. If no port is specified, TCP port 9001 is used by default. <i>Note: If this port is blocked or used by another process, the SIP Server will not start.</i>

2) Thirdparty Registered Database

Item	Default value	Explanation
On/Off	Off	Enable or disable to use the third party database system for Registered Database.
Registered Database URL		URL for the Registered Database (ex. jdbc:mysql://localhost/db)
Registered Database Driver		JDBC Driver for the Registered Database. (ex. com.mysql.jdbc.Driver)
User Name		User name for the Registered Database.
Password		Password for the Registered Database.

3) Thirdparty Users Database

Item	Default value	Explanation
On/Off	Off	Enable or disable to use the third party database system for Users Database.
Encrypt Users Passwords	true	Enable or disable the user password encryption.
Users Database URL		URL for the Users Database (ex. jdbc:mysql://localhost/db)
Users Database Driver		JDBC Driver for the Users Database. (ex. com.mysql.jdbc.Driver)
User Name		User name for the Users Database.
Password		Password for the Users Database.

4) Thirdparty Alias Database

Item	Default value	Explanation
On/Off	Off	Enable or disable to use the third party database system for Alias Database.
Alias Database URL		URL for the Alias Database (ex. jdbc:mysql://localhost/db)
Alias Database Driver		JDBC Driver for the Alias Database. (ex. com.mysql.jdbc.Driver)
User Name		User name for the Alias Database.
Password		Password for the Alias Database.

5) Radius

Item	Default value	Explanation
On/Off (Authentication)	Off	Enable or disable to use the Radius for Authentication.
Port Number (Authentication)	1812	Radius server port number for Authentication
Port Number (Accounting)	1813	Radius server port number for Accounting
Server IP Address		Radius server IP address
Shared Secret		Password for connecting to Radius server

3.8.5. Password

The Password page allows an administrator to change the login password for the Brekeke SIP Server Administration Tool. Administrator's default user id is "sa" and its password is "sa".

To set the password for authenticating SIP requests, please use **[User Authentication]** page and refer to the section "3.4. User Authentication".



3.8.6. Domains

The Domains page allows an administrator to manage multiple domains. With the Multiple Domains Mode, Brekeke SIP Server can host multiple domains on one server.

The buttons at the right side of each domain are for editing and deleting the domain.

Item	Default value	Explanation
Multiple Domains mode	Unchecked	When it is checked, the server can host multiple domains. While the Multiple Domains mode is enabled, an administrator of each domain can access the Brekeke SIP Server Administration Tool with their password.

Field Name	Explanation												
Domain	Name of the domain												
Authentication	Authentication policy												
	<table border="1"> <thead> <tr> <th>Policy</th> <th>Explanation</th> </tr> </thead> <tbody> <tr> <td>Realm</td> <td>This is set as the "realm" value. If left blank, the server's IP address is used as the "realm".</td> </tr> <tr> <td>REGISTER</td> <td>When set to "on", the Brekeke SIP Server authenticates REGISTER requests.</td> </tr> <tr> <td>INVITE</td> <td>When set to "on", the Brekeke SIP Server authenticates INVITE requests.</td> </tr> <tr> <td>Auth-user=user in To (REGISTER)</td> <td>When set to "yes", the Brekeke SIP Server will authenticate REGISTER requests only when authentication user name matches the user name in the To header. When set to "no", the Brekeke SIP Server will authenticate all REGISTER requests.</td> </tr> <tr> <td>Auth-user=user in From</td> <td>When set to "yes", the Brekeke SIP Server will authenticate requests only when authentication user name matches the user name in the From header. When set to "no", the Brekeke SIP Server will authenticate all requests.</td> </tr> </tbody> </table>	Policy	Explanation	Realm	This is set as the "realm" value. If left blank, the server's IP address is used as the "realm".	REGISTER	When set to "on", the Brekeke SIP Server authenticates REGISTER requests.	INVITE	When set to "on", the Brekeke SIP Server authenticates INVITE requests.	Auth-user=user in To (REGISTER)	When set to "yes", the Brekeke SIP Server will authenticate REGISTER requests only when authentication user name matches the user name in the To header. When set to "no", the Brekeke SIP Server will authenticate all REGISTER requests.	Auth-user=user in From	When set to "yes", the Brekeke SIP Server will authenticate requests only when authentication user name matches the user name in the From header. When set to "no", the Brekeke SIP Server will authenticate all requests.
	Policy	Explanation											
	Realm	This is set as the "realm" value. If left blank, the server's IP address is used as the "realm".											
	REGISTER	When set to "on", the Brekeke SIP Server authenticates REGISTER requests.											
	INVITE	When set to "on", the Brekeke SIP Server authenticates INVITE requests.											
Auth-user=user in To (REGISTER)	When set to "yes", the Brekeke SIP Server will authenticate REGISTER requests only when authentication user name matches the user name in the To header. When set to "no", the Brekeke SIP Server will authenticate all REGISTER requests.												
Auth-user=user in From	When set to "yes", the Brekeke SIP Server will authenticate requests only when authentication user name matches the user name in the From header. When set to "no", the Brekeke SIP Server will authenticate all requests.												

Button	Explanation
 Edit	Edit the domain. Please refer to "3.8.7. New Domain/Edit Domain".
 Delete	Delete the domain
New Domain	Add new domain. Please refer to "3.8.7. New Domain/Edit Domain".

3.8.7. New Domain / Edit Domain

New Domain page allows an administrator to add a new domain. Edit Domain page allows an administrator to modify the domain.

Item	Default value	Explanation
Domain		Name of the domain
Disabled	Unchecked	When it is checked, the domain is disabled.
Admin-Password		The password for the domain administrator to login to the Brekeke SIP Server Administration Tool.
Realm		This is set as the "realm" value. If left blank, the server's IP address is used as the "realm".

Authentication		
Item	Default value	Explanation
REGISTER	on	When set to "on", the Brekeke SIP Server authenticates REGISTER requests.
INVITE	off	When set to "on", the Brekeke SIP Server authenticates INVITE requests.
Auth-user=user in To (REGISTER)	on	When set to "on", the Brekeke SIP Server will authenticate REGISTER requests only when authentication user name matches the user name in the To header. When set to "off", the Brekeke SIP Server will authenticate all REGISTER requests.
Auth-user=user in From	off	When set to "on", the Brekeke SIP Server will authenticate requests only when authentication user name matches the user name in the From header. When set to "off", the Brekeke SIP Server will authenticate all requests.

3.8.8. Mirroring

The Mirroring feature requires two Brekeke SIP Server Advanced Editions called the Primary server and the Secondary server (as a backup server). With this feature, the Primary server can mirror its SIP session data and registration data to the Secondary server in real-time and the Secondary server can take over the service with the mirrored data if the Primary server goes down. Generally, the Mirroring feature is used with the Heartbeat feature which can detect failure of the Primary server and turns the Secondary server active. Please refer to the section "9. Mirroring/Heartbeat" for a general setting.

Note: The Mirroring feature is available in the Advanced Edition only.

1) Server Status

If the server is inactive, "INACTIVE" will be shown. If the Mirroring Mode is disabled even if the server is active, "Disabled" will be shown. Otherwise, the following information will be shown.

Field Name	Explanation
mirroring-role	Either "primary" or "secondary". The Primary server provides the service while the Secondary server receives mirrored data.
mirroring-address	It is the shared IP address between the Primary server and Secondary server. Users of the service need to use this IP address as a proxy and registrar.
mirroring-pair	These are the pair's IP addresses. For the Primary server, the secondary's IP address should be set. For the Secondary server, the primary's IP address should be set.

2) Mirroring Settings

Item	Default value	Explanation
On/Off	off	When set to "on", the Mirroring Mode is enabled.
Role	primary	When set to "primary", the server works as the Primary server. When set to "secondary", the server works as the Secondary server.
Virtual IP Address		It is the shared IP address between the Primary server and Secondary server. Users of the service need to use this IP address as a proxy and registrar. This IP address should be unique and accessible to users.
Pair IP Address		These are the pair's IP addresses. For the Primary server, the Secondary server's IP address should be set. For the Secondary server, the Primary server's IP address should be set.
Mirroring Request Pattern		This defines a packet pattern for mirroring. With this setting, the Primary server mirrors only specified packets to the Secondary server. The blank means any SIP packets. For example: When set to "INVITE CANCEL BYE", the Primary server mirrors only INVITE, CANCEL and BYE packets. When set to "!REGISTER", the Primary server will not mirror REGISTER packets.

Button	Explanation
Save and Apply	Save and apply changes
Save	Save changes.

3.8.9. Heartbeat

The Heartbeat feature provides a failover function. It monitors targets which are network entities such as other Brekeke SIP Servers at frequent intervals. When it detects a target is down, it executes pre-defined actions such as IP address switching or email notification.

An administrator can define multiple Heartbeat targets and actions here.

Since the feature uses ICMP to check the target's availability, a situation such as a physical port problem or a cable disconnection will trigger a failover. Also, please make sure that ICMP packets could be accepted at the firewall of target network entities.

For a general Mirroring deployment, the Heartbeat feature is required only on the Secondary server. Therefore, the firewall for the Primary server should accept ICMP packets sent from the Secondary server. To do this, an administrator may add the physical IP address of the Secondary server at the Primary server's firewall settings as a trusted IP address. Also, under the Mirroring deployment, please start the Primary server before starting the Heartbeat feature on the Secondary server. Please refer to the section "9. Mirroring/Heartbeat" for more information.

Note: The Heartbeat feature is available in the Advanced Edition only.

1) Heartbeat Status

This section shows current Heartbeat status and allows an administrator to start/stop the Heartbeat. The Heartbeat feature can start even if Brekeke SIP Server is inactive because it works independently from the server.

Field Name	Explanation
Status	If the Heartbeat feature is running, the status is "Running". If the Heartbeat feature is not running, the status is "Not Running". If the Heartbeat has failed, the status is "Failed".



Button	Explanation
Start	Start the Heartbeat feature <i>Note: Please start the Heartbeat feature manually after Brekeke SIP Server starts because it will not start automatically when the server starts.</i>
Stop	Stop the Heartbeat feature
Restart	Restart the Heartbeat feature
Latest log file	Download the latest Heartbeat log file if it is available. The previous log file will be removed when the Heartbeat feature starts.

2) Heartbeat Settings

This section shows current Heartbeat settings and its actions and allows an administrator to add/edit them. Multiple Heartbeats can be defined and each Heartbeat can have multiple actions. Changes take effect when the Heartbeat feature is restarted.

Please refer to the section “3.8.10. Heartbeat Settings” and to “3.8.11. Action Settings” for more details.

Field Name	Explanation										
Heartbeat	<table border="1"> <thead> <tr> <th>Field Name</th> <th>Explanation</th> </tr> </thead> <tbody> <tr> <td>IP Address</td> <td>IP address of the target entity.</td> </tr> <tr> <td>Timeout</td> <td>After the timeout period expires without any response from the target entity, specified actions will be executed. [milliseconds]</td> </tr> <tr> <td>Interval</td> <td>Broadcast interval for ICMP packet [milliseconds]</td> </tr> <tr> <td>Retry</td> <td>Maximum retries for ICMP packet</td> </tr> </tbody> </table>	Field Name	Explanation	IP Address	IP address of the target entity.	Timeout	After the timeout period expires without any response from the target entity, specified actions will be executed. [milliseconds]	Interval	Broadcast interval for ICMP packet [milliseconds]	Retry	Maximum retries for ICMP packet
	Field Name	Explanation									
	IP Address	IP address of the target entity.									
	Timeout	After the timeout period expires without any response from the target entity, specified actions will be executed. [milliseconds]									
	Interval	Broadcast interval for ICMP packet [milliseconds]									
Retry	Maximum retries for ICMP packet										
Action	This field displays information related to each action. These actions are executed when the Heartbeat feature detects a target is down										

Button	Explanation
 Edit	Edit the Heartbeat or action.
 Delete	Delete the action
New Heartbeat	Add new Heartbeat setting. Please refer to “3.8.10. Heartbeat Settings”
Add Action	Add new action setting. Please refer to “3.8.11. Action Settings”
Delete Heartbeat	Delete the Heartbeat setting.

3) Remote Access

The server accepts an action request only from the remote IP addresses defined in the IP Address Pattern. If this is undefined, the server accepts action requests from any remote IP address.

Field Name	Explanation
IP Address Pattern	Desired remote IP address pattern defined by Regular Expressions.

Button	Explanation
Save	Save changes.

3.8.10. Heartbeat Settings

Heartbeat Settings page allows an administrator to define and edit Heartbeat settings. Changes take effect when the Heartbeat feature is restarted.

Note: The Heartbeat feature is available in the Advanced Edition only.

Item	Default value	Explanation
IP Address		IP address of the target entity.
Timeout	3000	After the timeout period expires without any response from the target entity, specified actions will be executed. [milliseconds]
Interval	500	Broadcast interval for ICMP packet [milliseconds]
Retry	2	Maximum retries for ICMP packet

Button	Explanation
Save	Save changes and return to the previous page.
Cancel	Cancel changes and return to the previous page.

3.8.11. Action Settings

There are several action types which may be launched when the Heartbeat feature detects a target entity failure. Changes take effect when the Heartbeat feature is restarted.

Type	Explanation
Send Email	Send a notification e-mail to the specified e-mail address.
Re-initialize as primary	Re-initialize the server as the Primary server
Add IP Address (Linux/Win)	Add an interface IP address
Delete IP Address (Linux/Win)	Delete an interface IP address
Execute Command	Execute an external command
Management Command	Execute an internal server management command

1) Send Email

Send a notification e-mail to the specified e-mail address when the Heartbeat feature detects a target entity failure.

Item	Default value	Explanation
Type	Send Email	Send a notification e-mail to the specified e-mail address.
Position		The operation order
To		Receiver's e-mail address.
From		Sender's e-mail address.
Subject		E-mail subject
Body		E-mail body
SMTP Server		SMTP Server's address and port
POP3 Server		POP Server's address and port (If the SMTP server requires POP before SMTP.)
User		User Name
Password		Password
SMTP authentication	off	If the SMTP server requires an authentication, please set to "on".
Encrypted Connection (SSL)	off	If the SMTP server requires a SSL connection, please set to "on".

2) Re-initialize as primary

Re-initialize the Brekeke SIP Server as the Primary server when the Heartbeat feature detects a target entity failure. This action is used by the Secondary server when the original Primary server goes down.

Item	Default value	Explanation
Type	Re-initialize as primary	Re-initialize the server as the Primary server
Position		The operation order
Remote URL		The URL address of the desired server in which you want to execute the action. Please leave blank if the remote server is localhost.

3) Add IP Address

Add an interface IP address in the Brekeke SIP when the Heartbeat feature detects a target entity failure. Generally, this action is used to add the Virtual IP address defined in the Mirroring settings of the Secondary server when the original Primary server goes down.

Item	Default value	Explanation
Type	Add IP Address (Linux/Win)	Add an interface IP address
Position		The operation order
Interface Name		Name of the interface on the desired server which you want to execute the action. (for example "Local Area Connection", or "eth0")
IP Address		IP address
Subnet mask		Subnet Mask
Remote URL		The URL address of the desired server which you want to execute the action. Please leave blank if the remote server is localhost.

4) Delete IP Address

Delete an interface IP address from Brekeke SIP Server when the Heartbeat feature detects a target entity failure.

Item	Default value	Explanation
Type	Delete IP Address (Linux/Win)	Delete an interface IP address
Position		The operation order
Interface Name		Name of the interface on the desired server which you want to execute the action.
IP Address		IP address
Subnet mask		Subnet Mask
Remote URL		The URL address of the desired server in which you want to execute the action. Please leave blank if the remote server is localhost.

5) Execute Command

Execute an external command when the Heartbeat feature detects a target entity failure.

Item	Default value	Explanation
Type	Execute Command	Execute an external command
Position		The operation order
Command		Command name and its parameters
Remote URL		The URL address of the desired server in which you want to execute the action. Please leave blank if the remote server is localhost.

6) Management Command

Execute an internal server management command at the Brekeke SIP Server when the Heartbeat feature detects a target entity failure.

Item	Default value	Explanation
Type	Management Command	Execute an internal server management command
Position		The operation order
Target Address		
Parameters		Command name and its parameters
Text		
Remote URL		The URL address of the desired server in which you want to execute the action. Please leave blank if the remote server is localhost.

3.8.12. Advanced

The Advanced page allows an administrator to add/edit internal environment variables. Please refer to the section “10. Environment Variables” for reference.

3.9. Maintenance

The Maintenance, which is in the [Maintenance] menu of Brekeke SIP Server Admintool, is for performing backups, updating the software, and for activating the license.

3.9.1. Back Up

An administrator can back-up the existing settings using the Back Up option. The settings will be saved in the SST file.

3.9.2. Restore

With the Restore option, an administrator can restore the backup settings from the SST file.

3.9.3. Update Software

This page is for updating the Brekeke SIP Server. Please specify an update file (.war file) and push [Upload] button. After updating the software, please restart the computer.

3.9.4. Activate

This page is for activating the Brekeke SIP Server Product ID.

Note that if a same Product ID is used with multiple installations, the status of ID changes to Temporary.

4. Dial Plan

4.1. What is the Dial Plan?

The Brekeke SIP Server's Dial Plan defines rules for routing, filtering and actions. The Dial Plan can also be used for setting environment variables and modifications of selected SIP headers. Regular expressions are used for defining those rules.

This section is a reference for the Dial Plan functions. Please refer to the section "3.3. Dial Plan" for more details. For sample definitions, please refer to the "Brekeke SIP Server Tutorial-Dial Plan".

The Dial Plan can consist of multiple rules. Each rule is defined with the pair of Matching Patterns and Deploy Patterns. When all conditions set in the Matching Patterns are satisfied, the actions defined in the Deploy Patterns are applied.

By setting a Dial Plan, you can achieve the following functions:

- Routing
- Filtering
- Modifications of SIP headers
- Load Balancing
- RTP relay settings
- Call Session Plug-ins
- Call Dial Plan Plug-ins
- Setting Environment Variables

4.2. Create and Edit Dial Plan

To edit the Dial Plan, open **[Dial Plan]** menu. For creating new Dial plan rule, select **[New Rule]** option. For editing a current Dial Plan rule, click the corresponding edit icon. Please refer to the section "3.3.2. New Rule/ Edit Rule" for more details.

You can also edit Dial Plan files directly. Your changes will be in effect after you restart Brekeke SIP Server. The Dial Plan file is located under the install directly:

```
<INSTALL_DIR>\webapps\proxy\WEB-INF\work\sv\etc\dialplan.tbl
```

4.3. Matching Patterns

The Matching Patterns define conditions for applying the rule. Conditions can be defined using a pair of the following: the name of the SIP header, condition functions, system environment variables, source IP address, or the source port number, and the string pattern for matching. By defining multiple pairs, you can make the conditions more specific. Regular Expressions are used for defining string matching patterns. The string between parenthesis () in the right side can be referred to in Matching Patterns and Deploy Patterns.

To add a condition in the Matching Patterns section:

1. Push [...] button (which is between the Variable field and the Value field).
2. Select a variable name from the pull-down list or type a variable name directly in the Variable field.
3. Type a string pattern to the Value field and then, push the [+] button. Refer to the section “3.3.2. New Rule/ Edit Rule” for more information.

4.3.1. Syntax

Matching Patterns Syntax:

SIP_header_name = string pattern

&environment_variable_name = string pattern

\$condition_function_name = string pattern

\$condition_function_name(arguments) = string pattern

Main regular expressions which can be used in Matching Pattern’s right side are as follows:

Symbols	Meaning
!	If ‘!’ is placed in the front of the string pattern, it means NOT.
^	Match the beginning of the line
\$	Match the end of the line
[abc]	Match any character listed between brackets. In this case, a or b or c.
[^abc]	Match any character except those listed between the brackets. In this case, any characters except a, b and c.
.	Match any character except new line
X+	Match the preceding element (X, in this case) one or more times
X*	Match the preceding element (X, in this case) zero or more times
X{n}	Match the preceding element (X, in this case) n times

Symbols	Meaning
X{n,}	Match the preceding element (X, in this case) at least n times
X{n,m}	Match the preceding element (X, in this case) at least n times, but no more than m times
(chars)	The characters between the brackets will be put in a buffer. To refer to the n-th digit buffer in Deploy Pattern, use %<digit> (for example %1)

1) SIP Header Field Name

To use a SIP header as a condition, specify a pair of a SIP header name and a string pattern.

Syntax:

SIP header field name = a string pattern

Example:

From = sip:user@domain\.com[>]*

If the SIP URI in From: header is "sip:user@domain.com"

To = sip:11@

If the SIP user name in To: header field is "11"

To = sip:9(.+)@

If the SIP user name in To: header field starts with 9

To = sip:(....)@

If the SIP user name in To: header field contains only 4 characters

Supported = timer

If Supported: header field contains the string "timer",

Expires = ^[0-5]\$

If the value of Expires: header field is in the range 0-5

Contact = sip:[A-Za-z]+@

If the user name in Contact header contains only alphabet

2) Environment Variable

The environment variable is a variable name which starts with '&'. The variable name is not case sensitive. Please refer to the section "10. Environment Variables" for reference.

Syntax:

&variable_name = a string pattern

Example:

```
&sv.name = ^main-sv$
```

If the value of the server name (Environment variable: sv.name) is “main-sv”.

```
&net.sip.timeout.ringing = ^5[0-9][0-9][0-9]$
```

If the value of Ringing Timeout (Environment variable: net.sip.timeout.ringing) is in the range 5000-5999.

3) Conditional Function

The variable that starts with ‘\$’ is treated as a conditional function. The variable name is not case sensitive. Some conditional functions can have an argument.

If you want to create own conditional function, please refer to the “Dial Plan Plug-in Developer’s Guide”. The Built-in functions are described in section 4.3.2 below.

Syntax:

```
$conditional_function_name = a string pattern
```

```
$conditional_function_name(argument) = a string pattern
```

How to call functions:**Function_name(SIP header field name)**

If a SIP header field name is set as an argument to a conditional function, the value of the SIP header field will be passed to the function.

Example: \$func(From)

The value of From: header will be passed to the function “func”.

Function_name(&Environment_variable_name)

If an environment variable name is set as an argument to a conditional function, the corresponding value of the variable will be passed to the function. The prefix ‘&’ should be added before an environment variable name.

Example: \$func(&net.sip.timeout.ringing)

The value of environment variable net.sip.timeout.ringing will be passed to the function “func”.

Function_name(\$Conditional_function_name)

If a conditional function name is set as an argument to another conditional function, the return value of the “argument” function will be passed to the other conditional function. The prefix ‘\$’ should be added before a conditional function name.

Example: `$func1($func2)`

The return value of the function “func2” will be passed to the function “func1”.

Example: `$func1($func2($func3))`

The return value of the function “func3” will be passed to the function “func2” and the return value of the function “func2” will be passed to the function “func1”.

Example: `$func($func(To))`

The contents of To: header field will be passed to the function “func” and its return value will be passed to the function “func” again.

Function_name(“Text_String”)

If a text string is set as an argument, the text string is passed to the function. The text string should be enclosed in double quotes.

Example: `$func(“string”)`

The string “string” will be passed to the function “func”.

4.3.2. Reference of Conditional Functions

1) General Functions

\$addr

Meaning:

Source IP address of the incoming SIP packet

Syntax:

`$addr`

Explanation:

Returns the source IP address of the incoming request packet.

Example:

`$addr = ^127\.0\.0\.1$`

If the source IP address of the packet is the loopback address (127.0.0.1).

`$addr = ^192\.168\.`

If the source IP address of the packet starts with “192.168.”.

`$addr = ^172\.16\.0\.[1-5]$`

If the source IP address is in the range 172.16.0.1-172.16.0.5.

\$body**Meaning:**

Match in the message body

Syntax:

```
$body( regex )
      regex – regular expression
```

Explanation:

Gets the matched string from the message body such as SDP. The regular expression should contain a pair of brackets for defining the matched string.

Example:

```
$body( "m=audio (.+) RTP/AVP" ) = ^2000$
```

If the audio RTP port is 2000.

\$date**Meaning:**

Current Year/Month/Date

Syntax:

```
$date
$date( format )
      format – Date format
$date( format, timezone )
      format – Date format
      timezone – Time Zone
```

Explanation:

Returns the text string of current year/month/date. Date format can be specified as an argument. The default format is "yyyy/MM/dd".

Date format can consist of the following characters.

Character	Meaning	Character	Meaning
y	Year	m	Minute
M	Month	s	Second
d	Day	S	Millisecond
H	Hour		

Example:

```
$date = 2012/06/03
```

If the date is June 3rd, 2012.

```
$date = [15]$
```

If the last digit of the day is 1 or 5, i.e. the day of the month is 1,5,11,15, 21, 25, 31.

```
$date( "MM-dd-yyyy" ) = 06-03-2010
```

Gets the current date with the format "MM-dd-yyyy" and compares it with the string "06-03-2010".

```
$date( "YYYY/MM/dd", "JST" ) = (.+)
```

Gets the current date based on the time zone "JST".

\$geturi

Meaning:

Get the string of the SIP URI

Syntax:

```
$geturi( str )
```

str – text string

Explanation:

Gets the SIP URI part from the specified string.

Example:

```
$geturi( From ) = sip:user@domain\.com$
```

Gets the SIP URI from From header and compares with "sip:user@domain.com".

```
$geturi( $request ) = sip:1234@192\.168\.0\.1$
```

Gets the SIP URI part from the request-line (the return value of the conditional function "\$request") and compare it with the string "sip:1234@192.168.0.1".

\$globaladdr

Meaning:

If global address or not

Syntax:

```
$globaladdr( str )
```

str –IP address or FQDN

Explanation:

Checks if the address or FQDN specified as an argument is a global address or not.

If it is a global address, "true" will be returned. If not, "false" will be returned.

Example:

```
$globaladdr( "192.168.0.200" ) = false
```

If 192.168.0.200 is not a global address.

\$headerparam**Meaning:**

The header parameter

Syntax:

```
$headerparam( string )
```

str – string

```
$headerparam( string, key )
```

str – string

key – header parameter variable name

Explanation:

Returns the value of the header parameter variable from the specified string.

Example:

```
$headerparam( Contact ) = (.+)
```

Get all header parameters from Contact header.

```
$headerparam( To, "transport" ) = (.+)
```

Get the transport's value from To header's header parameters.

It is the same as `$param($headerparam(To), "transport")`.

\$istalking**Meaning:**

If talking or not

Syntax:

```
$istalking
```

```
$istalking( str )
```

str – SIP URI

Explanation:

Checks if the SIP URI specified as an argument is talking or not.

If it is talking, "true" will be returned. If not, "false" will be returned.

If no argument is set, Brekeke SIP Server checks if the Request URI is talking or not.

Example:

```
$istalking = true
```

If the Request URI is talking.

```
$istalking( "sip:user@192.168.0.2" ) = true
```

If the sip:user@192.168.0.2 is talking.

\$localhost

Meaning:

If localhost or not

Syntax:

```
$localhost
```

```
$localhost( str )
```

str – SIP URI or IP address or FQDN

Explanation:

Checks if the SIP URI or address specified as an argument is the localhost or not.

If it is localhost, “true” will be returned. If not, “false” will be returned.

If no argument is specified, Brekeke SIP Server checks if the source IP address of the packet is localhost or not.

Note: The addresses set in network interface settings in [Configuration] page will also be treated as “localhost”.

Example:

```
$localhost = true
```

If the source of the packet is localhost

```
$localhost( From ) = false
```

If the SIP URI in From header is not localhost

```
$localhost( "192.168.0.100" ) = true
```

If 192.168.0.100 is localhost

\$mirroring

Meaning:

If an incoming packet is a mirrored packet.

Syntax:

```
$mirroring
```

Explanation:

Checks if the incoming packet came from the primary SIP Server or not under the Mirroring mode. If an incoming packet is a mirrored packet, “true” will be returned. If not, “false” will be returned.

Note: This method is available in Advanced Edition only.

Example:

```
$mirroring = true
```

The incoming packet is a mirrored packet from the primary SIP Server.

\$mydomain

Meaning:

If my domain or not

Syntax:

```
$mydomain( str )  
str – domain name
```

Explanation:

Checks if the domain name specified as an argument is hosted by this server or not under the Multiple Domains mode. If it is my domain, “true” will be returned. If not, “false” will be returned.

The domain hosted by the server should be listed in the [Domains] page. Please refer to the section “3.8.6. Domains” for more details.

Example:

```
$mydomain( "sip.domain.com" ) = true  
If the “sip.domain.com” is hosted by this server.
```

\$not

Meaning:

Match in the message body

Syntax:

```
$not( value )  
value – true or false
```

Explanation:

If the value is "true", “false” will be returned. If not, “true” will be returned.

Example:

```
$not( $registered( "alice" ) ) = true  
If the user "alice" is not registered.
```

\$outbound

Meaning:

If outbound or not

Syntax:

```
$outbound  
$outbound( str )  
str – SIP URI or IP address or FQDN
```

Explanation:

Checks if the SIP URI or address set as an argument is outbound (IP address/port number which is not Brekeke SIP Server's IP address/port) or not.

If it is outbound, "true" will be returned. If not, "false" will be returned.

If no argument is set, Brekeke SIP Server checks if the Request URI is outbound or not.

For example, if Brekeke SIP Server's IP address is 192.168.0.1:5060, the IP address 192.168.0.2 or 192.168.0.1:6060 is considered as "outbound".

Example:

```
$outbound = true
```

If the Request URI contains an outbound address

```
$outbound( $request ) = true
```

If the Request URI contains an outbound address. (This is same as the case you didn't specify any argument.)

```
$outbound( To ) = false
```

If the SIP URI in To header is not outbound address.

```
$outbound ( "sip:user@host" ) = true
```

If "host" is outbound address.

\$param**Meaning:**

The parameter value

Syntax:

```
$param( str, key )
```

str – string

key – parameter variable name

Explanation:

Returns the value of the parameter variable from the specified string.

Example:

```
$param("sip:bob@192.168.0.1;expires=3600;q=1.0","expires")= ^300$
```

If the expires' s value is 300.

```
$param( Via, "branch" ) = (.+)
```

Get the branch' s value.

\$port**Meaning:**

Source port of the incoming SIP packet

Syntax:

```
$port
```

Explanation:

Returns the source port number of the incoming request packet.

Example:

```
$port = ^5060$
```

If the source port number of the packet is 5060.

```
$port = ^50[0-9][0-9]$
```

If the source port number of the packet is in the range 5000-5099.

\$primary**Meaning:**

If the server is the Primary server under the Mirroring mode.

Syntax:

```
$primary
```

Explanation:

Checks if the server is the Primary server or not under the Mirroring mode. If it is the primary, "true" will be returned. If not, "false" will be returned.

Note: This method is available in Advanced Edition only.

Example:

```
$primary = false
```

If the server is not Primary server under the Mirroring mode. It means the server is the Secondary server,

\$registered**Meaning:**

If registered or not

Syntax:

```
$registered
```

```
$registered( str )
```

str – SIP URI or a user name

Explanation:

Checks the SIP URI or the user name specified as an argument is registered in the Brekeke SIP Server's Registrar Database.

If the corresponding user is registered, "true" will be returned. If not, "false" is returned.

If no argument is specified, Brekeke SIP Server checks if the user in the Request URI is registered or not.

Example:

```
$registered = true
```

If the user in the Request URI is registered.

```
$registered( From ) = true
```

If the caller (The user in From header) is registered.

```
$registered( "alice" ) = false
```

If the user "alice" is not registered.

\$registeredaddr

See \$regaddr.

\$registereduri

See \$reguri.

\$regaddr**Meaning:**

The contact IP address of the registered user.

Syntax:

```
$regaddr
```

```
$regaddr( str )
```

str – SIP URI or a user name

Explanation:

Returns the contact IP address registered in the Registrar Database for the SIP URI or user name specified as an argument. If no argument is specified, the registered IP address for the user in the Request URI will be returned.

If any corresponding record can not be found, the condition will not be fulfilled.

Example:

```
$regaddr = ^192\.168\.0\.1$
```

If the user in the Request URI is registered from the IP address 192.168.0.1.

```
$regaddr( From ) = ^192\.168\.0\.200$
```

If the caller (the user in From header) is registered from the IP address 192.168.0.200.

```
$regaddr( "alice" ) = ^192\.168\.0\.
```

If the user "alice" registered from the IP address 192.168.0.x.

\$reguri

Meaning:

Contact SIP URI of the registered user.

Syntax:

```
$reguri
```

```
$reguri( str )
```

str – SIP URI or a user name

Explanation:

Returns the contact SIP URI registered in the Registrar Database for the SIP URI or user name specified as an argument. If no argument is specified, the registered contact SIP URI for the user in the Request URI will be returned.

If any corresponding user can not be found, this condition will not be fulfilled.

Example:

```
$reguri = sip:100@host
```

If the user's contact SIP URI of the request URI is "sip:100@host".

```
$reguri( "alice" ) = sip:admin@
```

If the user alice's contact SIP URI's user part is "admin".

\$request

Meaning:

SIP request Line

Syntax:

```
$request
```

Explanation:

Returns the SIP request line in the packet.

Example:

```
$request = sip:100@host
```

If the Request URI is "sip:100@host".

```
$request = ^INVITE
```

If the request is INVITE.

\$sid**Meaning:**

A session ID

Syntax:

`$sid`

Explanation:

Returns the session ID.

Session ID is a unique number assigned to each session.

Example:

`$sid = ^100$`

If the session ID is 100.

`$sid = [02468]$`

If the session id is an even number.

\$sessionnum**Meaning:**

The number of current sessions

Syntax:

`$sessionnum`

Explanation:

Returns the number of current sessions.

Example:

`$sessionnum = ^1000$`

If the number of current sessions reaches 1000.

\$soapget**Meaning:**

SOAP response

Syntax:

`$soapget(http-uri, namespace, method [,param [,param...]])`

http-uri – SOAP server's address

namespace– name space

method – method name

param – input parameter

Explanation:

Gets the information from the web service by SOAP.

Note: This method is available in Advanced Edition only.

Example:

```
$soapget("http://192.168.0.1","ns","get","in0=A","in1=B") = (.+)
```

\$subparam**Meaning:**

The subscriber parameter

Syntax:

```
$subparam( str )
```

str – string

```
$subparam( str, key )
```

str – string

key – subscriber parameter variable name

Explanation:

Returns the value of the subscriber parameter variable from the specified string.

Example:

```
$subparam( To ) = (.+)
```

Get all subscriber parameters from To header.

```
$subparam("sip:user;para=1@foo.com", "para" ) = (.+)
```

Get the "para" s value from the specified string.

It is the same as `$param($subparam("sip:user;para=1@foo.com"), "para")`.

\$time**Meaning:**

Current time

Syntax:

```
$time
```

```
$time( format )
```

format – Time format

```
$time( format, timezone )
```

format – Time format

timezone – Time Zone

Explanation:

Returns current time. Time format can be specified as an argument. The default format is "HH:mm:ss". For the details of the format, please refer to "\$date".

Example:

```
$time = 09:26:40
```

If the current time is 09:26:40.

```
$time = ^0[0-9]:
```

If the current time is from 0 to 9 o'clock.

```
$time( "SSSS" ) = [02468]$
```

If the millisecond is an even number.

```
$time( "HH:mm:ss", "PDT" ) = (.+)
```

Get the current time based on the time zone "PDT".

\$transport**Meaning:**

Transport type of the incoming SIP packet

Syntax:

```
$transport
```

Explanation:

Returns the transport type of the incoming request packet. A value will be "UDP" or "TCP".

Example:

```
$transport= ^UDP$
```

If the transport type is UDP.

```
$transport= ^TCP$
```

If the transport type is TCP.

\$uriparam**Meaning:**

The URI parameter

Syntax:

```
$uriparam( str )
```

str – string

```
$uriparam( str, key )
```

str – string

key – URI parameter variable name

Explanation:

Returns the value of the URI parameter variable from the specified string.

Example:

```
$uriparam( $request ) = (.+)
```

Get all URI parameters from the request URI.

```
$uriparam( To, "para" ) = (.+)
```

Get the “para”’s value from To header’s URI parameters.

It is the same as `$param($uriparam(To), "para")`

\$webget**Meaning:**

Match in the web page

Syntax:

```
$webget( http-uri, regex )
```

http-uri – website’s address

regex – regular expression

Explanation:

Gets the matched string from the specified web site. The regular expression should contain a pair of brackets for defining the matched string.

Note: This method is available in Advanced Edition only.

Example:

```
$webget( "http://www.foo.com/", "<B>(.)</B>" ) = (.+)
```

Get the string enclosed with and from the specified web site.

2) Alias Functions

The following functions allow an administrator to refer a record from Alias Database. Please refer the section “3.3.5. View Aliases” to configure and manage the database.

Note: The Alias feature is available in the Advanced Edition only.

\$alias.lookup**Meaning:**

Lookup from the Alias Database

Syntax:

```
$alias.lookup( alias_name )
```

alias_name – Alias Name

```
$alias.lookup( alias_name, group_id )
```

alias_name – Alias Name

group_id – Group ID

Explanation:

Returns corresponding entity value from the Alias Database for the Alias name specified as an argument.

Example:

```
$alias.lookup( "mike", "001" ) = (.+)
```

\$alias.reverse

Meaning:

Reverse lookup from the Alias Database

Syntax:

```
$alias.reverse( entity )
```

entity – Entity Name

```
$alias.reverse ( entity, group_id )
```

Entity – Entity Name

group_id – Group ID

Explanation:

Returns corresponding alias value from the Alias Database for the Entity name specified as an argument.

Example:

```
$alias.reverse ( "cell_phone" ) = (.+)
```

3) Mathematical Functions

The following functions allow an administrator to compare and manipulate numbers.

\$math.ge

Meaning:

Greater than or equal to (num1 <= num2)

Syntax:

```
$math.ge( num1, num2 )
```

Explanation:

If num1 is greater than or equal to num2, "true" will be returned. If not, "false" will be returned.

Example:

```
$math.ge( $sessionnum, "100" ) = true
```

If the number of current sessions is greater than or equal to 100.

\$math.gt**Meaning:**

Greater than ($\text{num1} > \text{num2}$)

Syntax:

```
$math.gt( num1, num2 )
```

Explanation:

If num1 is greater than num2, "true" will be returned. If not, "false" will be returned.

\$math.le**Meaning:**

Less than or equal to ($\text{num1} \leq \text{num2}$)

Syntax:

```
$math.le( num1, num2 )
```

Explanation:

If num1 is less than or equal to num2, "true" will be returned. If not, "false" will be returned.

\$math.lt**Meaning:**

Less than ($\text{num1} < \text{num2}$)

Syntax:

```
$math.lt( num1, num2 )
```

Explanation:

If num1 is less than num2, "true" will be returned. If not, "false" will be returned.

Example:

```
$math.lt( $sessionnum, "100" ) = true
```

If the number of current sessions is less than 100.

\$math.rand**Meaning:**

Random number

Syntax:

```
$math.rand( begin, end )
```

begin - beginning of the range

end - end of the range

Explanation:

Returns a random number from the specified range.

Example:

```
$math.rand( "2000", "3000" ) = (.+)
```

Get a random number from the range 2000-3000.

4) String Functions

The following functions allow an administrator to compare and manipulate strings.

\$str.equals**Meaning:**

Compares strings

Syntax:

```
$str.equals( str1, str2 [, str3] )
```

str– string

Explanation:

If specified strings are same, "true" will be returned. If not, "false" is returned.

Example:

```
$str.equals( $geturi(To), $geturi(From) ) = true
```

If the SIP URI of From header and To header are same.

```
$str.equals( $geturi(To), $geturi(From), $geturi(Contact) ) = true
```

If the SIP URI of From header, To header and Contact header are same.

\$str.hashcode**Meaning:**

Hash code

Syntax:

```
$str.hashcode( str )
```

str– string

Explanation:

Returns the hash codes of the specified string. The returning value is a hex string.

\$str.isdigits

Meaning:

If digits or not.

Syntax:

```
$str.isdigits( str )
```

str– string

Explanation:

If specified string is digits, “true” will be returned. If not, “false” is returned.

Example:

```
$str.isdigits( "1234" )= true
```

The string “1234” is digits.

\$str.length

Meaning:

Length of the string.

Syntax:

```
$str.length( str )
```

str– string

Explanation:

Returns the length of the specified string

\$str.md5

Meaning:

MD5 hash code

Syntax:

```
$str.md5( str )
```

str– string

Explanation:

Returns the MD5 hash codes of the specified string. The returning value is a hex string.

\$str.remove**Meaning:**

Removes a part of string

Syntax:

```
$str.remove( str1, str2 )  
str- string
```

Explanation:

Remove str2 from str1.

\$str.reverse**Meaning:**

Reverse string

Syntax:

```
$str.reverse( str )  
str- string
```

Explanation:

Reverse the specified string.

Example:

```
$str.reverse( "12345" )= (.+)  
Get the reversed string. It will be "54321".
```

\$str.substring**Meaning:**

Extracts a part of string

Syntax:

```
$str.substring( str, start )  
str- string  
start – Where to start the extraction  
$str.substring( str, start, end )  
str- string  
start – Where to start the extraction  
end – Where to stop the extraction
```

Explanation:

Extracts the characters in a string between two specified indices.

Example:

```
$str.substring( "sip:user@domain", 4 )= (.+)
```

Returns "user@domain".

```
$str.substring( "sip:user@domain", 4, 8 )= (.+)
```

Returns "user".

\$str.trim**Meaning:**

Trim string

Syntax:

```
$str.trim( str )
```

str- string

Explanation:

Strip whitespace from the beginning and end of the specified string.

Example:

```
$str.trim( " sip:user@domain " )= (.+)
```

Get the trimmed string. It will be "sip:user@domain".

5) User Directory Functions

The following function allows an administrator to refer a record from Users Database. Please refer the section "3.4. User Authentication" to manage the database.

\$usrdir.lookup**Meaning:**

Lookup from the Users Database

Syntax:

```
$usrdir.lookup
```

```
$usrdir.lookup( user_name )
```

user_name – User Name

Explanation:

Checks the user name specified as an argument is recorded in Users Database.

If the user is recorded, "true" will be returned. If not, "false" is returned.

If no argument is specified, Brekeke SIP Server checks if the authentication username in the SIP packet is recorded or not.

Example:

```
$usrdir.lookup( "mike" ) = true
```

If the user “mike” is recorded in the Users Database.

4.4. Deploy Patterns

The Deploy Patterns defines actions that will be taken when a rule's conditions defined in the Matching Pattern are met. At Deploy Patterns, you can define SIP header, routing destination, error response, environment variables as server's behavior, plug-in to load, and whether to perform RTP relay or not. Action is defined with a pair of SIP header name, handling variable name or environment variable and value. You can define multiple actions in one rule.

In the Value field of Deploy Patterns, matched string in Matching Patterns can be referred. When '%n' (n=number) was defined in value, the character string that locates in n-th number of parenthesis () in Matching Patterns's value will be inserted at the Deploy Patterns field.

To add a definition to the Deploy Patterns section:

1. Push [...] button (which is between the Variable field and the Value field).
2. Select a variable name from the pull-down list or type a variable name directly in the Variable field.
3. Type a value to the Value field and then, push the [+] button. Refer to the section “3.3.2. New Rule/ Edit Rule” for more information.

4.4.1. Syntax

Deploy Patterns Syntax:

SIP_header_field = setting value

&environment_variable_name = setting value

\$handling_variable_name = setting value

1) SIP Header Field Name

By specifying a SIP header name in variable field, you can replace, add or delete the value of the SIP header. If the specified SIP header field exists in a SIP packet, the header will be replaced with the specified value. If specified value is empty, the header will be removed from the SIP packet.

The SIP routing destination can be decided depending on the setting for “To” header as follows:

If **To = sip:username@host** is set, the SIP session will be routed to the address “host”.

If **To = sip:username@** is set, the SIP session will be routed to the contact address for the registered user “username” in Registrar Database.

Syntax:

SIP header field name = setting value

Example:

From = sip:admin@192.168.0.1

From header will be replaced with “sip:admin@192.168.0.1”.

To = sip:boss@192.168.0.100

To header will be replaced with “sip:boss@192.168.0.100”. And the session will be routed to the address “192.168.0.100”.

To = sip:sales@

The session will be routed to the contact address of the registered user “sales”.

From = "Ted" <sip:1650111@domain>

From header's SIP URI will be replaced with <sip:1650111@domain>. Caller's display name will be set as “Ted”.

Expires = 300

The value of Expires: will be set as 300.

User-Agent =

User-Agent: header will be deleted.

Refer-To = sip:user@server

Refer-To: header field will be replaced with “user@server”.

2) Environment Variable

The variable which starts with ‘&’ is treated as an environment variable. The environment variable name isn't case sensitive.

This setting will be applied only for the session that matches with Matching Patterns. To configure the environment variables for the whole system, please set them in the property file or in the Configuration page.

Syntax:

&environment_variable_name = a setting value

Example:

&net.sip.timeout.ringing = 10000

Set the value of ringing timeout to 10000.

(Set the environment variable `net.sip.timeout.ringing = 10000`)

&net.sip.addrecordroute = false

Don't add Record-Route: header.

(Set the environment variable `net.sip.addrecordroute = false`)

&net.rtp.audio.payloadtype = 0

Change the audio payload type in SDP to PCMU.

(Set the environment variable `net.rtp.audio.payloadtype = 0`)

3) Handling Variable

The variable which starts with '\$' is treated as a handling variable. Handling variables are not case sensitive.

Syntax:

`$handling_variable_name = a setting value`

4.4.2. Reference of Handling Variable

\$action

Meaning:

Command name to execute or response code to return

Syntax:

`$action = an internal command name`

`$action = a SIP response code`

Explanation:

If an internal command name is specified, the server executes the command.

If a SIP response code is specified, the server returns the response packet with the specified response code to the matched request and the session will not be routed.

Example:

`$action = 200`

Returns the response "200 OK".

\$auth**Meaning:**

Whether to authenticate or not

Syntax:

```
$auth = true or false
```

Explanation:

This sets whether to authenticate the matched request or not.

If "true", the authentication will be enabled. If "false", the authentication will be disabled.

The default value is the value which is set in **[Configuration]** page.

Example:

```
$auth = true
```

Authenticate the matched request

\$b2bua**Meaning:**

Whether to enable the B2B-UA (Back-To-Back User Agent) mode

Syntax:

```
$b2bua= true or false
```

Explanation:

If "true", the B2B-UA mode will be enabled. If "false", the B2B-UA mode will be disabled.

The default value is "false".

With the B2B-UA mode, Brekeke SIP Server hides Via and Record-Route headers and replaces original Call-ID header with a unique value.

Example:

```
$b2bua= true
```

B2B-UA mode is used for the session.

\$continue**Meaning:**

Whether Brekeke SIP Server continues checking the next rule or not.

Syntax:

```
$continue = true or false
```

Explanation:

This is a variable to make the server handle multiple rules.

If "true", Brekeke SIP Server continues to check the next rule below. If "false", Brekeke SIP

Server will not continue checking the next rules. The default is “false”.

As long as the Matching Patterns conditions are fulfilled and Deploy Patterns contains \$continue=true, Brekeke SIP Server continues checking rules.

Example:

```
$continue = true
```

Continues checking the next rule.

\$ifdst

Meaning:

Interface address used for sending/receiving packets to/from the session destination

Syntax:

```
$ifdst = IP address or FQDN
```

Explanation:

It is an interface address used for sending/receiving the packets to/from the session destination (UAS). This address is used for the values in Via, Record-Route headers.

Example:

```
$ifdst = 192.168.0.100
```

Set 192.168.0.100 as an interface address for the sending packets to the destination.

\$ifsrc

Meaning:

Interface address used for sending/receiving packets to/from the session originator

Syntax:

```
$ifsrc = IP address or FQDN
```

Explanation:

It is an interface address used for sending/receiving the packets to/from the session originator (UAC). This address is used for the values in Via, Record-Route headers.

Example:

```
$ifsrc = 192.168.1.200
```

Set 192.168.0.100 as an interface address for the sending packets to the session originator.

\$log**Meaning:**

Logging message

Syntax:

```
$log = a message
```

Explanation:

Brekeke SIP Server writes the specified message to the log file.

Example:

```
$log = debug:message
```

The server writes “debug:message” into the log file.

\$nat**Meaning:**

Whether to handle NAT traversal

Syntax:

```
$nat = true or false
```

Explanation:

If "true", the NAT traversal mode will be enabled. If “false”, the NAT traversal will be disabled. If "auto", Brekeke SIP Server will automatically decides whether to handle NAT traversal. The default value is "auto".

If the NAT traversal mode is enabled, RTP relay will also be enabled.

Example:

```
$nat = true
```

Handle NAT traversal.

\$replaceuri.from**Meaning:**

Whether to replace From header to appropriate SIP URI

Syntax:

```
$replaceuri.from = true or false
```

Explanation:

If "true", From header will be replaced with an appropriate SIP URI. If “false”, it is disabled.

If “auto”, Brekeke SIP Server will decide whether to replace the header or not automatically.

The default value is "auto".

Example:

```
$replaceuri.from = false
```

From header will not be replaced.

\$replaceuri.to**Meaning:**

Whether to replace To header to appropriate SIP URI

Syntax:

```
$replaceuri.to = true or false
```

Explanation:

If "true", To header will be replaced with an appropriate SIP URI. If "false", it is disabled. If "auto", Brekeke SIP Server will decide whether to replace the header or not automatically. The default value is "auto".

Example:

```
$replaceuri.to = false
```

To header will not be replaced.

\$request**Meaning:**

Request line

Syntax:

```
$request= a request line
```

Explanation:

Brekeke SIP Server replaces the request line of the matched request packet with the specified value.

Example:

```
$request= INVITE sip:201@domain SIP/2.0
```

Set "INVITE sip:201@domain SIP/2.0" as the new request line.

\$response**Meaning:**

Response code to return

Syntax:

```
$response = a SIP response code
```

Explanation:

The server returns the response packet with the specified response code to the matched request and the session will not be routed

Example:

```
$response = 400
```

Returns the response “400 Bad Request”.

\$rtp**Meaning:**

Whether to relay RTP packets

Syntax:

```
$rtp = true or false
```

Explanation:

If "true", RTP packets will be relayed through Brekeke SIP Server. If "false", RTP packets will not be relayed through Brekeke SIP Server. If “auto”, Brekeke SIP Server will decide whether to relay RTP packets or not automatically (For example, Brekeke SIP Server relays RTP packets for the UAs behind NAT). The default value is the value set in [Configuration] page.

Example:

```
$rtp = true
```

Enable RTP relay.

\$session**Meaning:**

Load a session plug-in.

Syntax:

```
$session = a session plug-in name
```

Explanation:

Specifies the name of session plug-in to use. For creating a plug-in, please refer to the “Session Plug-in Developer’s Guide”.

Example:

```
$session = com.sample.radius.proxy.RadiusAcct
```

Set the com.sample.radius.proxy.RadiusAcct class as a session plug-in.

\$target**Meaning:**

Routing destination

Syntax:

```
$target = IP address or FQDN
```

Explanation:

Sets the session's routing destination.

In the Advanced Edition, multiple destinations can be specified for failover. If the SIP Server can not reach a destination within an Inviting time-out, the next specified destination will be used.

Example:

```
$target = provider.domain
```

Routes the session to provider.domain.

```
$target = 172.16.10.1, 172.16.10.2, 172.16.10.3
```

Multiple destination IP addresses are specified for failover.

\$transport**Meaning:**

Transport type

Syntax:

```
$transport = a transport type ("UDP" or "TCP")
```

Explanation:

It is a transport type used for sending/receiving the packets to/from the session destination (UAS).

Example:

```
$transport = TCP
```

Use TCP for the sending packets to the UAS.

5. Upper Registration and Thru Registration

Using Upper Registration or thru Registration function, SIP clients under the Brekeke SIP Server will be registered at the other SIP Server (the Upper Server), and users can receive calls from the Upper Server through the Brekeke SIP Server.

5.1. Upper Registration

Upper Registration is a function to forward all of REGISTER requests to the registrar server (Upper Server) specified at the Brekeke SIP Server.

Please use the following settings for Upper Registration:

1. In the **[Configuration]** page > **[SIP]**, set **[Upper Registration]** as follows.

Item	Setting Value	Explanation
On/Off	on	Enable the Upper Registration
Registrar Server	The address of the other registrar server	IP address or FQDN of the registrar server to be used as the Upper Registration destination
Protocol	UDP or TCP	Transport protocol used for Upper Registration

2. Set the following at a SIP client.

Item	Setting Value
SIP Proxy Server	Brekeke SIP Server's IP address or FQDN
Registrar	Brekeke SIP Server's IP address or FQDN
Outbound Proxy	Brekeke SIP Server's IP address or FQDN
User Name	When authentication is set at the Upper Server, set the user name that is assigned by the Upper Server here.
Password	When authentication is set at the Upper Server, set the password that is assigned by the Upper Server here.

5.2. Thru Registration

Thru Registration is a function to forward REGISTER requests to the registrar server (Upper Server) specified in the Request-URI.

Please use the following settings for Thru Registration:

1. In the **[Configuration]** page > **[SIP]**, set **[Thru Registration]** as follows.

Item	Value	Explanation
On/Off	on	Enable the Thru Registration

2. Set the following at a SIP client.

Item	Setting Value
SIP Proxy Server	Brekeke SIP Server's IP address or FQDN In the case where "Outbound Proxy" setting is available, you would need to set the Upper Server's address here.
Registrar	Upper Server's address
Outbound Proxy	Brekeke SIP Server's IP address or FQDN
User Name	When authentication is set at the Upper Server, set the user name that is assigned by the Upper Server here.
Password	When authentication is set at the Upper Server, set the password that is assigned by the Upper Server here.

6. NAT Traversal

The NAT Traversal feature is used to keep connectivity when SIP clients are located on different networks. The feature rewrites SIP packets and relay RTP packets to meet requirements.

6.1. Brekeke SIP Server Behind NAT (Near-End NAT traversal)

If you are using the Brekeke SIP Server behind a NAT, but need to communicate with SIP servers and clients outside the NAT, please use the following settings for Near-End NAT traversal.

6.1.1. UPnP Settings

If your router supports UPnP, you can use it for Near-End NAT traversal. With UPnP, the Brekeke SIP Server can obtain the WAN IP address of the router and manage the port forwarding. Please use the following settings for UPnP:

1. Enable the UPnP at the router. (Refer to the router's document for more details.)
2. In the **[Configuration]** page > **[System]**, set **[UPnP]** as follows.

Item	Setting value	Explanation
Enable/Disable	enable	Enable the UPnP feature.
Default router IP address	The local IP address of the router	The local IP address of the router

3. Restart the Brekeke SIP Server.
4. Go to **[Server Status]** page. If the Brekeke SIP server got the WAN IP address of the router successfully, the IP address will be shown at the **[interface]** field. Also, the **[router]** field will show the router's information.

6.1.2. Manual Configuration

If your router doesn't support UPnP, you need to configure interface address settings and a port forwarding manually. Please use the following settings for manual configuration:

1. In the **[Configuration]** page > **[System]**, set **[Network]** as follows.

Item	Setting value	Explanation
Interface address 1	The WAN IP address of the router	IP address or FQDN of the router which can be reached from WAN side.

2. Setting port forwarding at the router is required to ensure NAT traversal to work properly.

With proper setting at the router, the Brekeke SIP Server's listening ports for SIP and RTP are forwarded to the Brekeke SIP Server's IP address.

Please set the following Brekeke SIP Server's ports for the port forwarding at the router.

Port Number (Default)	Transport Protocol	Purpose	Set at
5060	UDP and TCP	SIP	[Configuration] > [SIP] > [Local Port]
10000-10999	UDP	RTP	[Configuration] > [RTP] > [Minimum Port] and [Maximum Port]

6.2. For Clients Behind NAT over the Internet (Far-End NAT traversal)

To communicate properly with SIP clients located behind a NAT over the Internet, Far-End NAT traversal feature is applied to the call. If the Brekeke SIP Server is located behind a different NAT, you would need to set the Near-End NAT traversal setting as well.

1. Far-End NAT traversal requires maintaining port mapping at the router that is located at the same network with SIP clients. SIP packets from the Brekeke SIP Server will be undeliverable when port mapping has been cleared. To ensure maintaining the port mapping at the router, the Brekeke SIP Server needs to send dummy SIP packets for refreshing periodically; this feature is called Keep Alive. The interval of Keep Alive needs to be set short to prevent port mapping being cleared.

For some routers, this Keep Alive feature does not work to maintain port mapping. For such a case, we recommend that you use the Port Forwarding setting at the router instead. Please refer to the step.2.

In the **[Configuration]** page > **[SIP]**, set **[NAT traversal]** as follows.

Item	Setting Value	Explanation
Keep address/port mapping	on	Enable Keep Alive feature
Interval (ms)	(Depends on network environments.)	This is the interval to send dummy SIP packets. Default is set as 12,000 milliseconds (12 seconds). Shorter interval is recommended to ensure maintaining port mapping at routers.

2. In addition to the Keep Alive feature, there is another way to establish communications with a SIP client located behind a NAT over the Internet. When the communication cannot be established, even with Keep Alive settings, it is necessary to set port forwarding settings on the router located on the same network with SIP client. For port forwarding, you need to set the port numbers for SIP and RTP that SIP client is using on the router. Please refer to the configuration screen or document of SIP client for the port numbers to set at these settings.

7. Basic Setup

To have proper communications using the Brekeke SIP Server, precise settings at both Brekeke SIP Server and SIP client are necessary.

7.1. Setup Brekeke SIP Server

Generally, you may not need any settings at Brekeke SIP Server for basic setup. If you require the user authentication, please enable the Authentication feature and create users at the Brekeke SIP Server.

1. Enable the Authentication.

In the **[Configuration]** page > **[SIP]**, set **[Authentication]** as follows.

Item	Setting Value	Explanation
REGISTER	on	Authenticates REGISTER requests.
INVITE	on	Authenticates INVITE requests.

2. Create Users.

In the **[User Authentication]** page > **[New User]**, create a new user for authentication. Please refer to the section “3.4.2. New User / Edit User” for more details.

7.2. Setup SIP Client

Setting up the SIP client (User Agent, UA) begins with preparing an appropriate SIP client to meet your requirements and environment. Commonly used SIP clients are SIP softphones, SIP hardphones, VoIP Gateways, Analog Telephone Adaptor (ATA), and Instant Messenger (IM).

1. Basic settings for SIP clients. The setting items are depending on SIP clients.

Item	Setting Value
SIP Proxy Server	Brekeke SIP Server's IP address or FQDN
Registrar	Brekeke SIP Server's IP address or FQDN
Outbound Proxy	Brekeke SIP Server's IP address or FQDN
Domain	Brekeke SIP Server's IP address or FQDN
Realm	Brekeke SIP Server's IP address ✓ Set the same Realm which is set to the SIP Server if the server does authentication.

Item	Setting Value
User Name	User name.
Authentication User Name	When authentication is set at the Brekeke SIP Server, set the user name that is assigned by server.
Password	When authentication is set at the Brekeke SIP Server, set the password that is assigned by server.

2. If a SIP client is properly set, you can confirm registration status from the **[Registered Clients]** page on the Brekeke SIP Server Admintool.

7.3. Make a test call

After more than two SIP clients are registered in the Brekeke SIP Server, each SIP client can call each other by dialing a registered user name.

8. Security

This section describes how to configure Brekeke SIP Server security features. These features can protect your service against attacks or unauthorized use.

8.1. Administration Tool

To avoid a takeover of the server, please change the password for Administration Tool at **[Configuration] > [Password]** page. Its default password is "sa".

8.2. SIP Authentication

There are two ways to enable SIP Authentication. One is for the entire server. Another is for certain SIP requests.

To use SIP Authentication, an administrator needs to add users in the **[User Authentication]** page. Please refer to the section "3.4. User Authentication" for more details

8.2.1. SIP Authentication for all INVITE/REGISTER requests

Please enable SIP Authentication at the **[Configuration] > [SIP]** page. This setting affects all of INVITE / REGISTER requests.

Item	Setting Value	Explanation
REGISTER	on	Authenticates REGISTER requests.
INVITE	on	Authenticates INVITE requests.

8.2.2. SIP Authentication for certain requests

The server can authenticate a certain SIP request by using \$auth variable in the Dial Plan.

The \$auth variable determines whether to authenticate the matched request or not. If the value is "true", the server authenticates the request. If the value is "false", the server does not authenticate the request.

Example-1: Authenticate SUBSCRIBE requests.

Matching Patterns	Deploy Patterns
\$request = ^SUBSCRIBE	\$auth = true \$continue = true

Example-2: Don't authenticate INVITE requests if it comes from 192.168.0.x.

Matching Patterns	Deploy Patterns
<pre>\$request = ^INVITE \$addr = ^192.168.0</pre>	<pre>\$auth = false \$continue = true</pre>

8.3. To block a non-registered user's INVITE request.

To block non-registered users, the following sample Dial Plan rules will help you.

Example-1: If a client is not registered in the server, its INVITE request will be rejected with “403 Forbidden” response.

Matching Patterns	Deploy Patterns
<pre>\$request = ^INVITE \$registered(From) = false</pre>	<pre>\$action = 403</pre>

Example-2: If a client's registered IP address and port do not match with request's remote IP address and port, the request will be rejected with “403 Forbidden” response.

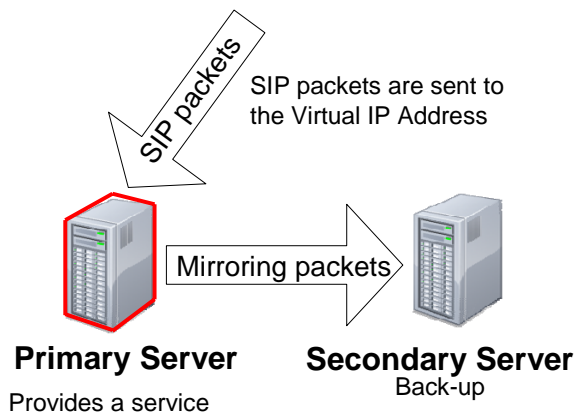
Matching Patterns	Deploy Patterns
<pre>\$request = ^INVITE \$port = (.+) \$addr = (.+) \$registeredaddr(From) != %1:%2</pre>	<pre>\$action = 403</pre>

9. Mirroring/Heartbeat

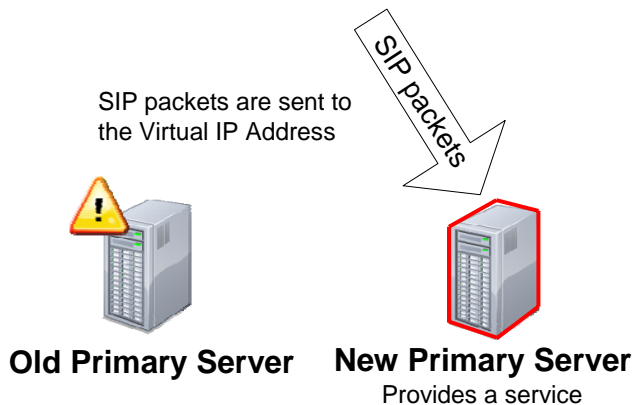
This section describes how to configure the Mirroring and Heartbeat features for a failover. The Mirroring and Heartbeat features provide High Availability (HA) functions for keeping your SIP service alive. To deploy these features, please prepare two server installations of the Brekeke SIP Server Advanced Edition called the Primary server and the Secondary server. With the Mirroring and Heartbeat features, third-parties clustering/failover solutions are no longer necessary.

9.1. Deployment Structure

1. Before the Primary server goes down, the Primary Server provides the service and the Secondary server stands by as an idle backup. All of SIP packets are sent to the Primary server with the Virtual IP address.



2. If the Primary server goes down, the Heartbeat feature switches the Secondary server to be the Primary server and assigns the Virtual IP address to the new Primary server.



9.2. The Primary Server Settings

9.2.1. Firewall Settings at the Primary Server

If there is a firewall on the Primary server, please configure it to accept ICMP packets sent from the Secondary server. For example, you can add the physical IP address of the Secondary server at the firewall settings as a trusted IP address.

9.2.2. Add the Virtual IP Address in the Primary Server

At the OS's network settings, please add a unique interface IP address as the Virtual IP Address.

9.2.3. Mirroring Settings at the Primary Server

Please configure the Mirroring settings at the **[Configuration]** > **[Mirroring]** page of the Primary Server. Then, please push **[Save]** button. Please refer to “3.8.8. Mirroring” for more details.

Item	Setting Value	Explanation
On/Off	on	Activates or deactivates the mirroring feature
Role	primary	Defines the role of this server.
Virtual IP Address		This is the shared IP address between the Primary server and Secondary server. Users of the service need to use this IP address as a proxy and registrar. This IP address should be unique and accessible to users.
Pair IP Address		The Secondary server's IP address

9.3. The Secondary Server Settings

9.3.1. Mirroring Settings at the Secondary Server

Please configure the Mirroring settings at the **[Configuration]** > **[Mirroring]** page of the Secondary Server. Then, please push **[Save]** button. Please refer to “3.8.8. Mirroring” for more details.

Item	Setting Value	Explanation
On/Off	on	Activates or deactivates the mirroring feature.
Role	secondary	Defines the role of this server.
Virtual IP Address		This is the shared IP address between the Primary server and Secondary server. Users of the service need to use this IP address as a proxy and registrar. This IP address should be the same IP address defined at the Primary server.
Pair IP Address		The Primary server's IP address

9.3.2. Heartbeat Settings for the Secondary Server

Please configure the Heartbeat settings at the **[Configuration]** > **[Heartbeat]** page of the Secondary Server. Please refer to "3.8.9. Heartbeat" for more details.

1. Please push **[New Heartbeat]** button.
2. Set the following at the **[Heartbeat Settings]** page.

Item	Explanation
IP Address	The Primary server's IP address
Timeout	After the timeout period expires without any response from the target entity, specified actions will be executed. [milliseconds]
Interval	Broadcast interval for ICMP packet [milliseconds]
Retry	Maximum retries for ICMP packet

3. Push the **[Save]** button.
4. Push the **[Add Action]** button.
5. Set the following first action.

Item	Setting Value	Explanation
Type	Add IP Address (Linux/Win)	
Position	1	The operation order
Interface Name		Name of the interface on the server on which you want to execute the action. (for example "Local Area Connection", or "eth0")
IP Address		The Virtual IP Address
Subnet mask		Subnet Mask

6. Push the **[Save]** button.
7. Push the **[Add Action]** button.
8. Set the following second action.

Item	Setting Value	Explanation
Type	Re-initialize as primary	
Position	2	The operation order

9. Push the **[Save]** button.

9.4. Start the Mirroring and Heartbeat features

Once the above settings are done, the administrator can start using the features.

9.4.1. Start the Primary Server

Please reboot the server machine and start the Primary server. The Mirroring feature will start automatically.

Note: Please start the Primary server before starting the Heartbeat feature on the Secondary server.

9.4.2. Start the Secondary Server

1. Please restart the Secondary server. The Mirroring feature will start automatically.
2. Push **[Start]** button at the **[Configuration] > [Heartbeat]** page. The Heartbeat feature will start.

Note: Please start the Heartbeat feature manually after Brekeke SIP Server starts because it will not start automatically when the server starts.

10. Environment Variables

Using of Environment Variables allows an administrator to tune behaviors of Brekeke SIP Server. They can be specified in Advanced page of the Configuration. Please refer to the section “3.8.10. Advanced”. Also, some of variables can be specified in the Dial Plan’s Deploy Patterns.

10.1. General

net.auth.accept.down

If true, accept SIP requests while the User Directory Database is down. (default: true)

net.bind.interface

The binding IP address.

net.dns.srv

If true, enable DNS SRV resolution. (default: true)

net.dns.srv.cache.size

The cache size for DNS SRV. (default: 32)

net.rtp.audio.payloadtype

The code of the payload type for RTP relay. If the value is specified, the Brekeke SIP Server specifies it in SDP.

net.sip.failover.dns.srv

If true, use DNS SRV for SIP failover (default: true)

net.sip.max.size

The maximum size of acceptable SIP packet. [bytes] (default: 65535)

net.sip.size

The buffer size for receiving a SIP packet over UDP. [bytes] (default: 65535)

10.2. Registrar

net.registrar.adjust.expires

The expiration value for adjusting. [seconds]

net.registrar.cache.size

The cache size for registrations. (default: 2000)

net.registrar.cache.use

If true, use the cache for registrations. (default: true)

net.registrar.maxtry

The maximum number of acceptable retries from a client. (default: 3)

net.registrar.min.expires

The minimum number of expiration for REGISTER. [seconds]

If the Expires of a received REGISTER is lower than the value, the response “423 Interval Too Brief” will be returned.

10.3. TCP

net.sip.tcp.max.connection

The maximum number of TCP connection. 0 means unlimited. (default: 0)

net.sip.tcp.keepalive.use

If true, use the TCP keep-alive feature with SO_KEEPALIVE. (default: true)

net.sip.tcp.reuse

If true, reuse a native TCP with SO_REUSEADDR. (default: true)

net.sip.tcp.timer.use

If true, close a TCP connection related with a REGISTER request when the registration is expired. (default: true)

net.sip.tcp.size.con.buffer

The buffer size for TCP connections. [bytes] (default: 8192)

net.sip.transport.follow.request

If true, a SIP packet will be sent over the same TCP connection of the initial request packet. (default: false)

10.4. UPnP

net.rtp.portmap.auto

If true, make the port-mapping for RTP automatically. (default: true)

net.upnp.multicast

If true, send a "discover" request to the multicast to find a router. (default: false)

net.upnp.timeout.retry.max

The maximum number for retrying a UPnP request. (default: 8)

net.upnp.timeout.retry.max

The maximum retries for a UPnP request. (default: 8)

net.upnp.timeout.retry.timer

The interval between retries of a UPnP request. [milliseconds] (default: 1000)

10.5. Logging

We recommend setting the following logging-level only for debugging. The higher logging level may reduce the server performance. The range of the logging level is 0 to 255. The default logging level is 0.

The log file will be stored under the following directly.

`<INSTALL_DIRECTORY>/webapps/proxy/WEB-INF/work/sv/log`

net.registrar.loglevel.file

The logging level of the Registrar. (default: 0)

net.sip.loglevel.file

The logging level of the Proxy. (default: 0)

net.tcp.loglevel.file

The logging level for TCP connections. (default: 0)

11. Uninstall

This section will assist you with uninstalling Brekeke SIP Server software from your computer.

11.1. Uninstall from Windows

Step 1: Stop Brekeke SIP Server

Push [Shutdown] button in the [Restart/Shutdown] page.

Step 2: Kill Java processes

Kill all Java processes (java.exe) used by Brekeke SIP Server from TaskManager.

Step 3: Execute the Uninstaller

Select "Uninstall Brekeke SIP Server" from Windows's Start menu.

11.2. Uninstall from Linux, Mac OS X, and Solaris

Step 1: Stop Brekeke SIP Server

Push [Shutdown] button in the [Restart/Shutdown] page.

Step 2: Stop Tomcat

Stop the Apache Tomcat. (e.g: `catalina.sh stop`)

Step 3: Remove files

Remove "proxy" folder and "proxy.war" file in Tomcat's webapps folder.

Appendix A: Glossary

- ◆ **Admintool, Administrative tool or Administration tool**

Front-end tool to manage Brekeke SIP Server. Because it is web-based, you can access the tool either locally or remotely. You can start/shutdown the server, check the server's status, and configure the environment.

 - Refer to the section "3. Brekeke SIP Server Administration Tool"

- ◆ **Client**

Software or a hardware used for starting/receiving a session. The client should support SIP protocol. For example, soft phones, IM clients, IP phones are clients. Brekeke SIP Server mediates the connection between those clients.

 - Refer to the section "7.2. Setup SIP Client"
 - Related words: Server, SIP, UA

- ◆ **Deploy Patterns or Action Patterns**

The patterns defined by you that determine the actions in Dial Plan. You can define to replace the SIP headers contents, to set the destination of a SIP packet, etc.

 - Refer to the section "3.3.2 Edit Rule", "4.4 Deploy Patterns"
 - Related words: Dial Plan, Rule, Matching Patterns

- ◆ **Dial Plan**

Dial Plan is one of the methods that Brekeke SIP Server uses to decide the routing destination of a session. Dial Plan can consist of multiple rules. Each rule is defined using pairs of Matching Patterns and Deploy Patterns. Only when the session matches with the conditions in Matching Patterns will the actions defined in Deployed Patterns be executed. You can view and edit the Dial Plan rules at Admintool > **[Dial Plan]** page. For the details, refer to the section "4. Dial Plan".

 - Refer to the sections "1.1. What is Brekeke SIP Server?", "3.5. Registered Clients", "6 Dial Plan"
 - Related words: Rule, Deploy Pattern, Matching Patterns

◆ Environment Variable

These are variables for setting Brekeke SIP Server's behavior and administration information, and various internal parameters. You can set the values of the environment variables in the property file or you can set some parts of those environment variables in the [Configuration] page.

To set different environment variables for each session, you need to specify it using Dial Plan's Deploy Patterns.

- Refer to the section "3.8. Configuration ", "4. Dial Plan"
- Related words: Deploy Pattern

◆ Far-End NAT traversal

NAT traversal of the UA (client) which is behind a NAT that is far from Brekeke SIP Server.

- Refer to the section "1.1. What is Brekeke SIP Server?", and the section "6. NAT Traversal".
- Related words: NAT traversal, Near-End NAT traversal.

◆ ITSP

Abbreviation of Internet Telephony Service Provider.

◆ Matching Patterns or conditions patterns

Conditions in Dial Plan rules. You can use regular expressions for defining conditions using SIP headers, source IP address of the packets.

- Refer to the section "4.3 Matching Patterns"
- Related words: Dial Plan, Rule, Deploy Patterns

◆ NAT (Network Address Translation) Traversal

When each client in the same session is behind a different NAT (firewall), Brekeke SIP Server connects those clients using its proprietary NAT traversal feature. RTP packets may be relayed through Brekeke SIP Server depending on the network environment. Brekeke SIP Server's NAT traversal features supports both Far-End NAT and Near-End NAT.

- Refer to the section "1.1 " hat is Brekeke SIP Server?", and the section "6. NAT Traversal".
- Related words: Near-End NAT traversal, Far-End NAT traversal, RTP relay

◆ Near-End NAT Traversal

NAT traversal of the UA (client) which is behind a NAT and which is in the same LAN as Brekeke SIP Server.

- Refer to the section “1.1. What is Brekeke SIP Server?”, and the section “6. NAT Traversal”.
- Related words: NAT traversal, Far-End NAT traversal

◆ Register database

The database that the client addresses are recorded based on the data in REGISTER requests sent from the clients. Brekeke SIP Server will look up the client’s registered address from the database for deciding the session’s routing destination, when needed. You can view the list of registered clients at Brekeke SIP Server admintool > [Registered List] page.

- Refer to the section “1.1. What is Brekeke SIP Server?”, and “3.5. Registered Clients”
- Related words: Thru Registration, Upper Registration

◆ RTP

Abbreviation of Real-time Transport Protocol. It is the protocol that clients use for sending/receiving media (voice, video, etc.). For the details, refer to RFC1889,1890.

- Refer to the section “3.8.4. RTP”
- Related words: SIP, RTP relay

◆ RTP relay or RTP tunnel

RTP packets are usually transmitted directly between clients (not through Brekeke SIP Server). But if it is difficult for those UAs to directly communicate with each other depending on the network environment, Brekeke SIP Server will relay RTP packets. Brekeke SIP Server uses the port 10000-10999 (by default) for RTP relay.

- Refer to sections “1.1. What is Brekeke SIP Server?”, and “3.8.3. RTP”
- Related words: NAT traversal, RTP

◆ Rule or Dial Plan rule

A rule is a pair of Matching Patterns and Deploy Patterns for setting Dial Plan.

- Refer to the section “6. Dial Plan”
- Related words: Dial Plan, Deploy Patterns, Matching Patterns

◆ Session

A session is initiated by an INVITE request. For the voice conversation, 1 session is usually used for a call. A session remains until a BYE request is processed or an error response is processed. Sessions status can be checked at admintool > **[Session List]** page.

- Refer to section “3.6. Active Sessions”
- Related words: SIP

◆ Session ID or SID

A unique id assigned for each session.

- Refer to section “3.6. Active Sessions”
- Related words: Session

◆ Server

Server means Brekeke SIP Server in this document unless otherwise noted.

- Refer to section “1.1. What is Brekeke SIP Server?”

◆ SIP

Abbreviation of Session Initiation Protocol. It is a protocol that clients and servers use for setting up sessions or for controlling calls, etc. For the details, refer to RFC3261.

Brekeke SIP Server will send SIP packets sent from a client to an appropriate destination. The Server edits the SIP packets before sending to the destination as needed. The Server uses the port number 5060 (by default) for SIP.

- Refer to section “1.1. What is Brekeke SIP Server?”
- Related words: RTP, Session, Server, Client

◆ Thru Registration

If the Request URI in the REGISTER request sent from a client doesn't include Brekeke SIP Server's address, Brekeke SIP Server will forward the REGISTER request to the address specified in the request URI.

- Refer to sections “1.1. What is the Brekeke SIP Server?”, and “5.2. Thru Registration”
- Related words: Register database, Upper registration

◆ UA or User Agent

- Related words: Client

- ◆ User directory database

The database that holds the records of user information such as user name, password, etc. for authenticating SIP requests. You can view and edit the user information at Admintool > [Authentication] page.

To authenticate users using Brekeke SIP Server, user information needs to be added to the user directory database in advance.

- Refer to section "3.4. User Authentication".

- ◆ Upper Registration

This feature forwards REGISTER requests sent from clients to another server as configured at Brekeke SIP Server. A client can send just one REGISTER request to Brekeke SIP Server to register itself both at Brekeke SIP Server and at other server.

- Refer to sections "1.1. What is Brekeke SIP Server?", and "5.1. Upper Registration"
- Related words: Register database, Thru registration