

Brekeke SIP Server

Version 2.x

アドミニストレーター・ガイド

Brekeke Software, Inc.

バージョン

Brekeke SIP Server v2.x アドミニストレーター・ガイド (2007 年 4 月)

著作権

本ドキュメントの著作権は Brekeke Software, Inc. にあります。

Copyright ©2002 - 2007 Brekeke Software, Inc.

本ドキュメントの一部または全部を、Brekeke Software, Inc.との書面による同意なしに、複製、複製、転載、他言語への翻訳、書き換えをすることは法律で禁じられています。

免責事項

Brekeke Software, Inc.は予告なしに本ドキュメントの内容を変更する権利を有します。

登録商標

- ◆ Linux は Linus Torvalds 氏の米国及びその他の国における登録商標あるいは商標です。
- ◆ Red Hat は米国 Red Hat, Inc. の登録商標です。
- ◆ Windows は米国 Microsoft Corporation の米国及びその他の国における登録商標です。
- ◆ Mac は米国及びその他の国における Apple Computer, Inc の登録商標です。
- ◆ Java 及び Java のロゴは Sun Microsystems, Inc.の米国及びその他の国における登録商標あるいは商標です。
- ◆ その他の製品名と会社名は、一般にその会社の登録商標です。

1.	はじめに	5
1.1.	Brekeke SIP Server とは?	5
2.	動作環境	7
3.	インストール (WINDOWS 版).....	8
3.1.	ステップ 1: Java SE SDK のインストール.....	8
3.2.	ステップ 2: Brekeke SIP Server のインストール.....	8
3.3.	ステップ 3: Brekeke SIP Server HTTP サービスの起動.....	8
3.4.	ステップ 4: Brekeke SIP Server 管理ツールの起動	8
4.	インストール (インストーラーを使用しない場合)	10
4.1.	ステップ 1: Java SE のインストール	10
4.2.	ステップ 2: Tomcat のインストール	10
4.3.	ステップ 3: Brekeke SIP Server のインストール.....	10
4.4.	ステップ 4: Tomcat の起動	10
4.5.	ステップ 5: Brekeke SIP Server 管理ツールの起動.....	10
5.	管理ツール	12
5.1.	Server Status ページ	12
5.2.	Restart / Shutdown ページ.....	13
5.3.	Dial Plan ページ	14
5.4.	User Authentication ページ	17
5.5.	Registered Clients ページ	20
5.6.	Active Sessions ページ.....	23

5.7.	Call Log ページ	26
5.8.	Configuration ページ	27
6.	ダイヤル・プラン	38
6.1.	ダイヤル・プランとは?	38
6.2.	編集	38
6.3.	文法	39
7.	UPPER REGISTRATION	58
8.	THRU REGISTRATION	59
9.	NATトラバーサル	60
9.1.	Near-End NATトラバーサル	60
9.2.	Far-End NATトラバーサル	61
10.	クライアントの設定	62
10.1.	クライアントの用意	62
10.2.	設定項目	62
10.3.	レジスタの確認	63
Appendix . A	用語解説	64

1. はじめに

本ドキュメントでは Brekeke SIP Server のインストールと各種設定について説明をしています。
必要に応じて以下のドキュメントも合わせて参照してください。

ダイヤルプランの設定について:

- Brekeke SIP Server チュートリアル - ダイヤル・プラン編

プラグインの開発について:

- Brekeke SIP Server Accounting Plug-in Developers Guide (英文)
- Brekeke SIP Server Authentication Plug-in Developer's Guide (英文)
- Brekeke SIP Server Dial Plan Plug-in Developer's Guide (英文)

1.1. Brekeke SIP Server とは？

Brekeke SIP Server は SIP プロキシ、レジストラ、NATトラバーサル、認証、コール・ルーティングなどの機能を提供する SIP サーバです。Brekeke SIP Server を利用することによって、SIP に対応したソフトフォン、ハードフォン、VoIP ゲートウェイ及びアプリケーションのあいだにおいて相互にコミュニケーションを行うことができます。

以下は Brekeke SIP Server が提供する主な機能です。

◆ コール・ルーティング

Brekeke SIP Server は クライアント(または UA、以下クライアント)や他のサーバなどから受信した SIP パケットを最適なルーティング先に転送します。ルーティング先にはレジスタ・データベースに登録されている SIP URI が利用されますが、ダイヤル・プランの定義を適用させることもできます。これらにより、ユーザーは着信者の最終的な SIP URI が判らなくても、Brekeke SIP Server でルーティング先が解決できればセッションを確立することができます。

ダイヤル・プランでは SIP パケットの各ヘッダや IP アドレスを正規表現でマッチングおよびフィルタリングできるため、柔軟にルーティング先を解決することができます。たとえばダイヤル・プランを使うことによって、オフィスが複数ある環境においては電話番号のプリフィックスでルーティング先のオフィスを選別することなどが可能になります。

◆ レジストラ・サービス

Brekeke SIP Server はクライアントから登録リクエスト(REGISTER リクエスト)を受け取り、レジスタ・データベースを更新します。このとき REGISTER リクエストで指定されている SIP URI

がユーザーのアドレスとしてレジスタ・データベースに登録されます。これにより、クライアントのユーザーは各自のユニークな SIP URI を用いて、他のユーザーから着信を受けることができます。

◆ **NATトラバース**

発信者と着信者が異なるネットワークにいる場合、Brekeke SIP Server は NATトラバース機能によって双方から送出された SIP パケットを最適な情報に書き換えて転送します。一般的に LAN 内ではプライベート・アドレスが使われているため、グローバル・アドレスのユーザーと通信をするために NATトラバース機能が必要となります。また、状況に応じて RTP パケットをリレー(中継) し、音声などのメディアが不達になることを防ぎます。

Brekeke SIP Server の NATトラバース機能は、サーバとクライアントが同じファイアーウォール内に存在する場合(Near-End NAT)と、クライアントが遠隔のファイアーウォール内に存在する場合(Far-End NAT)の両方に対応しています。

◆ **Upper/Thru レジストレーション**

Upper/Thru レジストレーション機能は、Brekeke SIP Server に対して送られた REGISTER リクエストをユーザーが指定した他の SIP サーバに転送する機能です。これによりユーザーは他の SIP サーバに対しても同時にレジストをすることができます。

たとえば、この機能によってユーザーは LAN 内にあるクライアントを Brekeke SIP Server を介して IP 電話事業者、(ITSP) のサーバに登録し、プロバイダの他のユーザーや公衆回線網上の電話機から着信を受けることが可能になります。

2. 動作環境

Brekeke SIP Server は以下のプラットフォームで動作します。

OS	Microsoft Windows, Linux, Mac OS X, Solaris
Java	JDK 1.4 以降
推奨メモリ容量	256 MB 以上

- ✓ Brekeke SIP Server を Windows 以外の OS にインストールする場合は、あらかじめ Apache Tomcat (v4.1.2 以降) をインストールする必要があります。
- ✓ Brekeke SIP Server を管理するために、別途ウェブブラウザをご用意ください。

3. インストール (Windows 版)

3.1. ステップ 1: Java SE SDK のインストール

Java Standard Edition (SE) を入手し、対象とする環境にインストールします。

Java SE は以下のウェブページからダウンロードが可能です。Windows に対応した JDK または JRE のバージョン 1.4 以降をご利用下さい。

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

3.2. ステップ 2: Brekeke SIP Server のインストール

- 1) Brekeke Software, Inc. からインストーラーを入手してください。同時に電子メールでプロダクト ID が送付されます。
- 2) インストーラーを起動します。
- 3) インストーラーの指示に従ってインストールを進めてください。インストーラーの最後のページで**[Run Brekeke SIP Server]** にチェックをし、**[Finish]** ボタンを押すと自動的に Brekeke SIP Server の HTTP サービスが起動します。

3.3. ステップ 3: Brekeke SIP Server HTTP サービスの起動

インストーラーの最後で**[Run Brekeke SIP Server]** をチェックしなかった場合は、以下のいずれかの方法で Brekeke SIP Server HTTP サービスを起動してください。

- **[スタート]**→**[コントロールパネル]**→**[管理ツール]**→**[サービス]** を開き、**[Brekeke SIP Server]** のサービスを選択して起動します。
- Windows を再起動します。自動的に Brekeke SIP Server HTTP サービスが起動されます。

3.4. ステップ 4: Brekeke SIP Server 管理ツールの起動

- 1) **[スタート]**→**[プログラム]**→**[Brekeke]**→**[Brekeke SIP Server]**→**[Brekeke SIP Server Admintool]** を選択します。ライセンス許諾 (License Agreement) ページが表示されますので、電子メールによって送付されたプロダクト ID を入力し **[I agree]** をクリックします。そして **[Activate]** を押してライセンスをアクティベートします。
- ✓ プロダクト ID の入力が必要となるのは、Brekeke SIP Server を新規にインストールする場合 または OnDO SIP Server から Brekeke SIP Server にアップグレードする場合のみです。
 - ✓ Brekeke SIP Server 以降のマイナーアップグレードの場合にはプロダクト ID のアクティベートは必要ありません。

- 2) 管理ツールのログイン画面で、ユーザーID とパスワードを入力し、**[Login]**ボタンをクリックします。
 - ✓ デフォルトの管理者用のアカウント名とパスワードは User ID: sa Password: sa です。

- 3) ログイン後に、Status が **“active”** と表示されていれば Brekeke SIP Server は正常に起動しています。Status が **“inactive”** の場合は正常に起動していません。このとき起動エラーの内容が Status の上に表示されます。
 - ✓ 他のアプリケーションが Brekeke SIP Server と同じポートを使用している場合などには、正常に起動できないため、Status に"inactive" と表示されます。その場合は他のアプリケーションを終了させて管理ツールの[Start/Shutdown]ページにある[Start]ボタンを押してください。Brekeke SIP Server はデフォルト設定において、ポート 5060 を SIP パケットの送受に使用します。

4. インストール (インストーラーを使用しない場合)

4.1. ステップ 1: Java SE のインストール

Java Standard Edition (SE) を入手し、対象とする環境にインストールします。

Java SE は以下のウェブページからダウンロードが可能です。動作 OS に対応した JDK または JRE のバージョン 1.4 以降をご利用下さい。

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

Java SE のインストール後に、環境変数 JAVA_HOME に J2SE のインストール・ディレクトリを設定してください。

4.2. ステップ 2: Tomcat のインストール

Tomcat を入手しインストールします。

Tomcat は以下のウェブページからダウンロードが可能です。動作 OS に対応した Tomcat のバージョン 4.1.2 以降のバイナリ版をダウンロードしてください。

<http://jakarta.apache.org/tomcat/>

4.3. ステップ 3: Brekeke SIP Server のインストール

Brekeke SIP Server のアーカイブ・ファイル(zip 形式)に含まれている war 形式のファイルを Tomcat 配下の webapps ディレクトリにコピーしてください。

4.4. ステップ 4: Tomcat の起動

- 1) Tomcat を起動します。
- 2) ウェブブラウザで <http://localhost:8080> にアクセスします。
(Tomcat のインストール時に、ポート番号として 8080 以外を指定した場合は、該当するポート番号にアクセスしてください。)
- 3) Apache Jakarta Project のページが表示されれば、Tomcat は正常に動作しています。

4.5. ステップ 5: Brekeke SIP Server 管理ツールの起動

- 1) ウェブブラウザで <http://localhost:8080/proxy/> にアクセスします。
(Tomcat のインストール時に、ポート番号として 8080 以外を指定した場合は、該当するポート番号にアクセスしてください。)
Brekeke SIP Server 管理ツールのログイン・ページが表示されます。User ID と Password を入力し、**[Login]** ボタンを押してログインします。
✓ デフォルトの管理者用のアカウント名とパスワードは User ID: sa Password: sa です。

- 2) ライセンス許諾 (License Agreement) ページが表示されますので、電子メールによって送付されたプロダクト ID を入力し **[I agree]** をクリックします。そして **[Activate]** を押してライセンスをアクティベートします。

- 3) 管理ツールのログイン画面で、ユーザーID とパスワードを入力し、**[Login]** ボタンをクリックします。
 - ✓ デフォルトの管理者用のアカウント名とパスワードは User ID: sa Password: sa です。

- 4) ログイン後に、Status が “**active**” と表示されていれば Brekeke SIP Server は正常に起動しています。Status が “**inactive**” の場合は正常に起動していません。このとき起動エラーの内容が Status の上に表示されます。
 - ✓ 他のアプリケーションが Brekeke SIP Server と同じポートを使用している場合などには、正常に起動できないため、Status に “inactive” と表示されます。その場合は他のアプリケーションを終了させて管理ツールの [Start/Shutdown] ページにある [Start] ボタンを押してください。Brekeke SIP Server はデフォルト設定において、ポート 5060 を SIP パケットの送受に使用します。

5. 管理ツール

5.1. Server Status ページ

管理ツールのメニューから[SERVER STATUS] ボタンをクリックすることによって表示されます。

このページではサーバの情報と状態を表示します。

Server Status	
項目名	説明
Status	サーバの状態 (例. ACTIVE, INACTIVE)
server-product	プロダクト名
server-ver	サーバのバージョン
server-name	サーバの名称 ^(*1)
server-description	サーバの説明 ^(*1)
server-location	サーバの設置場所 ^(*1)
server-startup-time	サーバの起動時刻
server-life-length	サーバの動作時間
machine-name	サーバが動作するホストの名前 (ホスト名)
listen-port	サーバが SIP パケットの送受に使用しているポート番号 ^(*1)
interface	サーバが使用するネットワーク・インターフェース・アドレス ^(*1)
startup-user	サーバのサービスまたは Tomcat を起動したユーザー名 (OS 情報から取得)
work-directory	サーバが稼動しているディレクトリ
session-active	現在のセッション数
session-total	サーバ起動後に処理されたセッション数のトータル
command-active	現在の管理コマンドの処理数
command-total	サーバ起動後に処理された管理コマンド数のトータル
sip-packet-total	サーバ起動後に受信した SIP パケットのトータル
registered-record	レジスタ・データベースに登録されているレコード数
os-name	OS の名称
os-ver	OS のバージョン
java-ver	Java のバージョン
admin-sip	管理者の SIP URI ^(*1)
admin-mail	管理者のメールアドレス ^(*1)

✓ (*1) [Configuration]ページで変更可能。「5.8 Configuration ページ」を参照してください。

Database Status	
項目名	説明
registered-database	レジスタ・データベースの接続状況
userdir-database	ユーザー・ディレクトリ・データベースの接続状況

5.2. Restart / Shutdown ページ

管理ツールのメニューから[Restart/Shutdown] ボタンをクリックすることによって表示されます。

このページでは Brekeke SIP Server の起動と停止を行います。

サーバが動作中であれば Status に”active”と表示し、停止中であれば “inactive” と表示します。

ボタン	説明
Start	サーバを起動します。
Restart	サーバを再起動します。
Shutdown	サーバを停止します。 処理中のセッションが残っている場合は、シャットダウンをするかどうかの確認メッセージを表示します。確認後に [Force Shutdown] ボタンを押した場合、すべてのセッションが終了されます。

5.3. Dial Plan ページ

5.3.1. Dial Plan ページ

[Dial Plan] の [View Rules] サブメニューではダイヤル・プランのルールを一覧表示します。またルールの優先度を変更することもできます。

一覧は優先度の高いルールから昇順で表示されますが、無効状態のルールについてはグレーで表示されます。ルールの編集、コピー、削除は、各ルールの右に位置するボタンによって行います。また、ページ下部の[Apply Rules]ボタンを押すことによって、サーバの動作中であったとしても変更された最新の内容を適用することができます。

項目名	説明
Priority	ダイヤルプラン・ルールの優先度
Name	ルール名
Matching Patterns	適用条件
Deploy Patterns	処理内容

ボタン	説明
Edit	ルールを編集します。
Copy	ルールをコピーします。
Delete	ルールを削除します。
Apply Rules	編集内容を保存し適用します。

5.3.2. ルールの新規追加・編集

[New Rule] サブメニューで 新規ダイヤルプランを作成できます。また **[Edit Rule]** サブメニューでダイヤルプランの変更ができます。

ダイヤル・プランの記述方法については、「6. ダイヤル・プラン」を参照してください。

項目名	デフォルト値	説明
Rule name		ルール名
Description		ルールの説明
Priority		ダイヤルプランの優先度
Disabled	チェックなし	無効化の有無 (チェックがある場合は、無効になります。)
Matching Patterns		Matching Pattern(条件)の一覧
Deploy Patterns		Deploy Pattern(処理内容)の一覧
Variable		変数名 [...]ボタンを押すことによって、主な変数が [Variable] で選択可能になります。
Value		Matching Patterns の場合: 変数を評価する値 Deploy Patterns の場合: 変数に代入する値

ボタン	説明
Insert	[Variable] と [Value] に記載した定義を一覧に挿入します。
Delete	選択されている定義を削除します。 削除された定義は、 [Variable] と [Value] に表示されます。
Down	選択されている定義を下に移動します。
Up	選択されている定義を上に移動します。
Save	編集内容を保存し、 [View Rules] ページに戻ります。
Cancel	編集内容を破棄し、 [View Rules] ページに戻ります。

5.3.3. ルールのインポート

[Import Rules] サブメニューで新規ダイヤルプランをアップロードしてインポートすることができます。

[参照] ボタンを押して、インポートするファイルを選択し、**[Upload]** ボタンをクリックしてファイルをアップロードします。

5.3.4. ルールのエクスポート

[Export Rules] サブメニューを選択することにより、現在のダイヤルプランのルールをファイルとしてエクスポートすることができます。

5.4. User Authentication ページ

管理ツールのメニューから[User Authentication]をクリックすることによって表示されます。

このページでは認証のためのユーザーの登録・編集を行います。

サーバの認証機能を有効にする場合は、「5.8.2. SIP(General)」を参照し、[Configuration]ページで設定を行ってください。

管理ツールにアクセスするためのパスワードの変更は、「5.8.6 Login Password」を参照し、[Configuration]ページで行ってください。

5.4.1. View Users

[View Users] サブメニューを選択すると、認証用のユーザーリストが表示されます。ここでは、ユーザの検索、編集、削除 が可能です。

ユーザ・ディレクトリ・データベースからユーザを検索します。[Pattern] や [On Field] フィールドでフィルターを設定して [Search] ボタンをクリックすることにより、特定のユーザ情報を検索することができます。

フィルタ

フィルター・タイプ	デフォルト値	説明
Pattern		検索するキーワード
On Field	User	どのフィールド内を検索するかを指定します。選択肢: User、Name、Email Address、Description
Maximum Rows	100	表示する行数

ボタン	説明
Search	検索を実行

5.4.2. New User

ユーザー認証情報を編集します。

(* = 必須設定項目)

	項目名	デフォルト値	説明
*	User		認証用ユーザー名
	Password		パスワード
	(Confirm)		パスワード再入力(確認用)
	Name		ユーザーの名前
	Email Address		ユーザーのメールアドレス
	Description		その他のユーザー情報

ボタン	説明
Modify	編集内容の更新
Add	編集内容の追加
New	新規編集のために全項目をクリア
Delete	編集内容のユーザーを削除

5.4.3. Import Users

[User Authentication] ページで [Import Users] をクリックすることによって表示されます。

複数ユーザーのユーザー情報をインポートして一括登録することができます。

ファイルは CSV (カンマ区切り) 形式です。1 ユーザーにつき以下のような構成の 1 行を記述します。

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

必要に応じてチェックボックスをチェックし、[Browse] ボタンを押してインポートする CSV 形式のファイルを指定します。最後に [Upload] ボタンを押すことによって、ファイルからユーザー認証情報が読み込まれます。

項目名	デフォルト値	説明
Overwrite existing records.	チェックなし	既存のレコードを上書きします。
Delete all existing records.	チェックなし	すべての既存のレコードを削除します。

ボタン	説明
Browse (または参照)	ファイルの選択
Upload	ファイルのアップロード

5.4.4. Export Users

エクスポート機能を使用して認証ユーザの情報をファイルに保存できます。

[Export Users] サブメニューをクリックすることにより、ファイルにレコードをエクスポートします。

5.5. Registered Clients ページ

管理ツールのメニューから[Registered Clients]をクリックすることによって表示されます。

5.5.1. View Clients

このページではサーバのレジスタ・データベースに登録されているレコードの一覧を表示します。
レコードは、クライアントから送信された REGISTER リクエストをサーバが処理することによって自動的に追加または更新されます。

フィルタ

項目名	デフォルト値	説明
Pattern		検索するキーワード
On Field	User	どのフィールド内を検索するかを指定します。

ボタン	説明
Search	検索を実行

レジスタ・レコードのリスト

項目名	デフォルト値	説明
User		ユーザーが他のクライアントからコンタクトを受けるときの名前です。
Contact URI		サーバがユーザーにコンタクトするための SIP URI です。
Expires	3600 (秒)	レジスタ・データベースでレコードを保持する期間です。保持期間が過ぎたレコードはレジスタ・データベースから自動的に削除されます。 レジスタ・データベースに保持されている間は、他のクライアントからコンタクトを受けることができます。 単位は秒。デフォルトは 3600 秒 (=1 時間)です。
Priority	1000	サーバがクライアントにコンタクトする際の優先順位です。 設定値の範囲は 100 ~ 1000。
Requester		このレコードの登録元 IP アドレスです。
Time Update		このレコードの最終更新時間です。

ボタン	説明
Unregister	該当するレコードをレジスタ・データベースから削除します。
Refresh	表示内容を更新します。

5.5.2. New Client

このページでは新規レコードを手動で追加できます。

項目名	デフォルト値	説明
User		ユーザーが他のクライアントからコンタクトを受けるときの名前です。
Contact URI		サーバがユーザーにコンタクトするための SIP URI です。
Expires	3600 (秒)	レジスタ・データベースでレコードを保持する期間です。保持期間が過ぎたレコードはレジスタ・データベースから自動的に削除されます。 レジスタ・データベースに保持されている間は、他のクライアントからコンタクトを受けることができます。 単位は秒。デフォルトは 3600 秒 (=1 時間)です。
Priority	1000	サーバがクライアントにコンタクトする際の優先順位です。 設定値の範囲は 100 ~ 1000。

ボタン	説明
Register	レジストラ・データベースに新規レコードをレジスタします。

5.6. Active Sessions ページ

5.6.1. 現在存在するセッションの一覧表示

管理ツールのメニューから[Active Sessions]をクリックすることによって表示されます。

このページでは現在のセッションの一覧を表示します。任意のセッションの[Session ID] をクリックすることによって選択されたセッションの詳細な情報を表示します。

フィルタ

項目名	説明
From	発信者の SIP URI または IP アドレス
To	ルーティング先の SIP URI または IP アドレス
Time Range	検索する時間の範囲
Status	セッションの状態

セッションのリスト

項目名	説明																
Session ID	セッション ID (クリックすることによって詳細な情報を表示します。)																
From	発信者の SIP URI と IP アドレス																
To	ルーティング先の SIP URI と IP アドレス																
Time	セッションの開始時刻																
Status	セッションの状態 <table border="1" data-bbox="588 1364 1054 1749"> <thead> <tr> <th>状態</th> <th>説明</th> </tr> </thead> <tbody> <tr> <td>Initializing</td> <td>初期化</td> </tr> <tr> <td>Inviting</td> <td>リクエスト送信</td> </tr> <tr> <td>Provisioning</td> <td>セッション確立準備</td> </tr> <tr> <td>Ringing</td> <td>リングング</td> </tr> <tr> <td>Accepted</td> <td>セッション確立</td> </tr> <tr> <td>Talking</td> <td>会話中</td> </tr> <tr> <td>Closing</td> <td>セッション終了</td> </tr> </tbody> </table>	状態	説明	Initializing	初期化	Inviting	リクエスト送信	Provisioning	セッション確立準備	Ringing	リングング	Accepted	セッション確立	Talking	会話中	Closing	セッション終了
状態	説明																
Initializing	初期化																
Inviting	リクエスト送信																
Provisioning	セッション確立準備																
Ringing	リングング																
Accepted	セッション確立																
Talking	会話中																
Closing	セッション終了																

リンク	説明
Refresh	表示内容を更新します。

5.6.2. Session Detail ページ

[Session List]ページで [Session ID]をクリックすることによって表示されます。

このページでは各セッションの詳細な情報を表示します。

項目名	説明																
From-url	発信者の SIP URI																
From-ip	発信者の IP アドレス																
From-if	発信者側のネットワーク・インターフェース																
To-url	ルーティング先の SIP URI																
To-ip	ルーティング先の IP アドレス																
To-if	ルーティング先側のネットワーク・インターフェース																
Call-id	Call-ID																
rule	適用されているダイヤル・プランのルール																
plugin	適用されているセッション・プラグイン																
sip-packet-total	受信した SIP パケットのトータル																
Listen-port	サーバが SIP パケットの送受に使用しているポート番号																
sip-packet-stacked	サーバで未処理の SIP パケット数																
Session-status	セッションの状態 <table border="1" data-bbox="588 1196 1054 1581"> <thead> <tr> <th>状態</th> <th>説明</th> </tr> </thead> <tbody> <tr> <td>Initializing</td> <td>初期化</td> </tr> <tr> <td>Inviting</td> <td>リクエスト送信</td> </tr> <tr> <td>Provisioning</td> <td>セッション確立準備</td> </tr> <tr> <td>Ringling</td> <td>リングング</td> </tr> <tr> <td>Accepted</td> <td>セッション確立</td> </tr> <tr> <td>Talking</td> <td>会話中</td> </tr> <tr> <td>Closing</td> <td>セッション終了</td> </tr> </tbody> </table>	状態	説明	Initializing	初期化	Inviting	リクエスト送信	Provisioning	セッション確立準備	Ringling	リングング	Accepted	セッション確立	Talking	会話中	Closing	セッション終了
状態	説明																
Initializing	初期化																
Inviting	リクエスト送信																
Provisioning	セッション確立準備																
Ringling	リングング																
Accepted	セッション確立																
Talking	会話中																
Closing	セッション終了																
time-inviting	セッション開始時刻																
time-talking	会話開始時刻																
length-talking	会話時間																
rtp-relay	RTP リレーの有無																

[rtp-relay]が on である場合は以下の情報が表示されます。

これは[rtp-srcdst] (発信者→着信者)と [rtp-dstsrc] (着信者→発信者) の各 RTP ストリームの状態を表します。

項目名	説明
media	メディア (audio=音声、video=動画)
transport	メディアのタイプ
payload	ペイロード・タイプ
status	状態 (active=ストリーミング中、hold=停止中)
listen-port	サーバにおける RTP パケットの受信ポート
send-port	サーバにおける RTP パケットの送信ポート
target	RTP パケットの送信先
packet-count	パケット数
packet/sec	1 秒あたりのパケット数
current size	最新の RTP のパケットサイズ (バイト)
buffer size	バッファサイズ (バイト)

ボタン	説明
Refresh	表示内容を更新します。
Disconnect	セッションを終了させます。
Back	[Session List]ページに戻ります。

5.7. Call Log ページ

管理ツールのメニューから[Call Log]をクリックすることによって表示されます。

このページでは日付毎のセッション数の一覧を表示します。また、任意のセッション数をクリックすることによって、該当日の接続履歴を表示します。

項目名	説明
HTML	セッション数をクリックしたとき、該当日の接続履歴を別ウィンドウで表示します。
CSV	セッション数をクリックしたとき、該当日の接続履歴を CSV で取得します。

ボタン	説明
Save	ログの保持期間を指定します。保持期間を過ぎたログは自動的に削除されます。

5.7.1. セッション履歴ウィンドウ

[Log]ページで、セッション数をクリックすることによって表示されます。

このウィンドウでは該当日の ログを表示します。

項目名	説明
sid	セッション ID
from-url	発信者の SIP URI
to-url	ルーティング先の SIP URI
talking-length	会話時間
invite-start-time	セッション開始時刻
talk-start-time	会話開始時刻
end-time	セッション終了時刻
result	結果
error	エラーコード (SIP のレスポンス番号、正常終了は -1)

ボタン	説明
Close	ウィンドウを閉じます。

5.8. Configuration ページ

管理ツールのメニューから[Configuration]をクリックすることによって表示されます。

このページでは環境設定、及びソフトウェアのアップグレードを行います。

設定した内容は、サーバを再起動することによって有効になります。

5.8.1. System

サーバの管理者情報やネットワークに関する設定をします。

これらの情報は[Server Status]ページでも表示されます。

(* =必須設定項目)

General			
	項目名	デフォルト値	説明
	Server Name	your-sip-sv	サーバの名称です。
	Server Description	your SIP Server	サーバの説明です。
	Server Location	your-place	サーバの設置場所です。
	Administrator SIP URI	your-sip-uri	管理者の SIP URI です。
	Administrator Email Address		管理者のメールアドレスです。
*	Start up	Auto	“auto” であれば、ウェブサーバ(Tomcat)起動時に、自動的に Brekeke SIP Server も起動します。 選択肢は “manual”, “auto”

Network			
	項目名	デフォルト値	説明
	Interface address 1-5		<p>サーバがインターフェースとして利用する IP アドレスまたはドメイン名 (FQDN) を指定します。</p> <p>インターフェースとして利用可能な IP アドレスは、サーバがインストールされたホストに割り当てられている IP アドレスです。</p> <p>OS が Windows の場合、サーバは自動的にホストの IP アドレスを取得するため、それらを指定する必要はありません。</p> <p>サーバが NAT 内にある場合は、NAT のグローバル IP アドレスまたはその FQDN も指定してください。また、IP アドレスが動的に変化する環境においては、一意な FQDN をインターフェースとして指定することによって、IP アドレスの変化に対応することができます。</p>
*	DNS caching period	3600	<p>DNS によるネーム解決の結果を保持する期間です。</p> <p>-1 の場合は、DNS キャッシュの更新はされません。</p> <p>単位は秒です。</p>
*	Auto interface discovery	off	<p>“on” であれば、自動的にインターフェース・アドレスを更新します。選択肢は “on”, “off”</p>
*	IP address filter	disable	<p>“allow” の場合、SIP Server は、[Filter Pattern] フィールド内に指定された IP アドレスのみから SIP パケットを受け付けます。</p> <p>“block” の場合、SIP Server は、[Filter Pattern] フィールドに指定された IP アドレス以外の IP アドレスから送られた SIP パケットを、すべて受け付けます。</p>
*	Filter Pattern		<p>正規表現で IP アドレスのパターンを指定します。</p>
*	Multiple Domains	no	<p>“yes” の場合、SIP Server で複数のドメインを使用でき、複数のドメインを判別してセッションを処理します。</p>

Java			
	項目名	デフォルト値	説明
	Java VM arguments		<p>Java VM に渡すパラメータです。</p> <p>但し、classpath パラメータは指定できません。</p>

ボタン	説明
Save	設定内容を保存します。 保存した内容はサーバを再起動することで有効になります。

5.8.2. SIP (General)

SIP 一般、NATトラバーサル及び認証に関する設定をします。

NAT トラバーサルの詳細は「9. NAT トラバーサル」を参照してください。認証用ユーザー・アカウントの登録・編集は「5.4 User Authentication ページ」を参照し、**[User Authentication]**ページで行ってください。

(* = 必須設定項目)

SIP exchanger			
	項目名	デフォルト値	説明
*	Session Limit	-1	サーバで同時に処理する最大セッション数です。 無制限の場合は -1 を指定します。
*	Local Port	5060	サーバが SIP パケットを送受するポート番号です。 特に理由が無い限り、5060 を指定してください。

NAT traversal			
	項目名	デフォルト値	説明
*	Keep address/port mapping	on	“on” であれば、Far-End NAT 内のクライアントに対して一定間隔でパケットを送ることにより、NAT における使用ポートのマッピングを維持(キープ・アライブ)します。 選択肢は “on”, “off”
	Interval	12000(ミリ秒)	キープ・アライブ用パケットの送信間隔を指定します 単位はミリ秒。デフォルトは 12000 ミリ秒(=2 分)です。
	Add 'rport' (Send) parameter	on	“on” の場合、サーバは Via ヘッダに rport パラメータをつけ INVITE を送ります。これにより、サーバ自身のグローバルのポート番号を知ることができます。
	Add 'rport' (Receive) parameter	off	“on” の場合、サーバは Via ヘッダに INVITE 送信元のポート番号を rport に設定して Via ヘッダに追加し、レスポンス(たとえば 100 Trying, 180 Ringing, 200 OK)を返します。

Authentication			
	項目名	デフォルト値	説明
	REGISTER	off	“on” であれば、REGISTER リクエストを認証します。
	INVITE	off	“on” であれば、INVITE リクエストを認証します。
	Realm		realm 文字列
	Auth-user=user in “To:” (Register)	no	“yes” であれば、認証用のユーザー名と REGISTER リクエストの To:ヘッダのユーザー名が一致する場合のみ認証します。
	Auth-user=user in “From:” (Invite)	no	“yes” であれば、認証用のユーザー名と INVITE リクエストの From:ヘッダのユーザー名が一致する場合のみ認証します。
	FQDN only	no	“yes” であれば、リクエストの SIP URI が FQDN を含む場合のみ認証します。SIP URI が IP アドレスを含む場合は認証しません。

ボタン	説明
Save	設定内容を保存します。 保存した内容はサーバを再起動することで有効になります。

5.8.3. SIP (Advanced)

Upper Registration 及び Thru Registration 機能やタイムアウトに関する設定をします。

Upper Registration については「7. Upper Registration」、Thru Registration については「8. Thru Registration」をそれぞれ参照してください。

(* = 必須設定項目)

Upper Registration			
	項目名	デフォルト値	説明
	On/Off	off	“on” であれば、Upper Registration 機能を有効にします。
	Register Server		Upper Registration 先のレジスタ・サーバの IP アドレスまたは FQDN です。

Thru Registration			
	項目名	デフォルト値	説明
	On/Off	off	“on” であれば、Thru Registration 機能を有効にします。

Timeout (無制限の場合は 0 を設定します。)			
	項目名	デフォルト値	説明
*	Ringing Timeout	120000 (ミリ秒)	呼び出し(Ringing)のタイムアウト時間です。
*	Talking Timeout	259200000 (ミリ秒)	会話のタイムアウト時間です。
*	Upper/Thru Timeout	30000 (ミリ秒)	Upper registration 及び Thru registration のタイムアウト時間です。

Miscellaneous			
	項目名	デフォルト値	説明
	100 Trying	any requests	“any requests” の場合、どのリクエストに対しても毎回 100 Trying を返します。

ボタン	説明
Save	設定内容を保存します。 保存した内容はサーバを再起動することで有効になります。

5.8.4. RTP

RTP の取り扱いに関する設定をします。

Brekeke SIP Server は、セッションが NAT を介する場合(例えば、発信者と着信者が異なるネットワークにいる場合)、RTP パケットを確実に各クライアントに到達させるために RTP パケットのリレー(中継)を行うことができます。

(* =必須設定項目)

RTP exchanger			
	項目名	デフォルト値	説明
*	RTP relay	auto	<p>“on” であれば、サーバ経由で RTP パケットをリレーします。</p> <p>“auto” であれば、サーバが自動的に判断します。(セッションが NAT を介する場合は、RTP パケットがリレーされます。)</p> <p>選択肢は “on”, “auto”</p>
*	RTP relay (UA on this machine)	auto	<p>“auto” であれば、サーバが自動的に判断します。</p> <p>“off” であれば、サーバと同じホストで稼働するクライアントに対してはサーバ経由で RTP パケットをリレーしません。</p> <p>選択肢は “auto”, “off”</p>
*	Minimum Port	10000	RTP パケットをリレーするポート範囲の最小値です。
*	Maximum Port	10999	RTP パケットをリレーするポート範囲の最大値です。
*	Minimum Port (Video)	0	<p>動画ストリーム用パケットをリレーするポート範囲の最小値です。</p> <p>0 の場合は[Minimum Port]の設定値が使われず。</p>
*	Maximum Port (Video)	0	<p>動画ストリーム用パケットをリレーするポート範囲の最大値です。</p> <p>0 の場合は[Maximum Port]の設定値が使われず。</p>
*	Port mapping	source port	<p>Far-End-NAT 内のクライアントへ RTP パケットを送信する場合の宛先ポート番号の取得方法です。</p> <p>“sdp” であれば、SIP パケットの SDP で指定されたポート番号を使います。</p> <p>“source port” であれば、RTP パケットの送信元ポートを宛先として使います。</p> <p>選択肢は “sdp”, “source port”</p>

Timeout (0=unlimited) (無制限の場合は 0 を設定します。)			
	項目名	デフォルト値	説明
*	RTP Session Timeout	600000 (ミリ秒)	サーバ経由で RTP パケットをリレーしているとき、RTP パケットが検出されなくなってからのタイムアウト時間です。 単位はミリ秒。デフォルトは 600000 ミリ秒 (=10 分)です。

ボタン	説明
Save	設定内容を保存します。 保存した内容はサーバを再起動することで有効になります。

5.8.5. Database

データベースに関する設定を行います。

Embedded Database			
	項目名	デフォルト値	説明
	Port Number		Brekeke SIP Server に組み込まれたデータベースが使用するポート番号 何も指定されていない場合は、デフォルトで TCP9001 が使用されます。

サード・パーティーのデータベースを使用する場合

Brekeke SIP Server の内部データベース(Embedded Database)を使用する代わりに、サードパーティーのデータベースを使用することができます。ここでは例として MySQL を使用する場合を説明します。

1. MySQL の JDBC ドライバー "MySQL Connector/J" を <http://www.mysql.com/products/connector/j/> から入手します。
2. JDBC ドライバーのファイル "mysql-connector-java-X.X.XX-XX-bin.jar" を <proxy_install_dir>\webapps\proxy\WEB-INF\lib にインストールします。
3. 次に示すように、データベース (例. mydb) を作成し 2 つのテーブルを作成します。
 - 1) MySQL のコマンドウィンドウを開き、新しいデータベース (例: mydb) を作成してください。

```
mysql> create database mydb;
```
 - 2) 以下のように t_registered テーブルを作成しインデックスを作成してください。
これはユーザ・エージェントのレジスタ情報を保存するためのテーブルです。

```
CREATE TABLE t_registered
(
    skey          BIGINT,
    namealias     VARCHAR(300),
    nameoriginal  VARCHAR(300),
    urlalias      VARCHAR(300),
    urloriginal   VARCHAR(300),
    acceptpattern VARCHAR(300),
    requester     VARCHAR(100),
    expires       BIGINT,
    priority      INT,
    timeupdate    BIGINT,
    expirestime   BIGINT,
    mappedport    VARCHAR(100),
    awake         INT,
    useragent     VARCHAR(300),
    param         VARCHAR(300)
);
```

```
CREATE INDEX idx_registered_skey ON t_registered( skey );
CREATE INDEX idx_registered_urlalias ON t_registered( urlalias );
CREATE INDEX idx_registered_urloriginal ON t_registered( urloriginal );
CREATE INDEX idx_registered_expirestime ON t_registered( expirestime );
```

3) 以下のように t_userdir テーブルとそのインデックスを作成してください。
これはユーザの認証用のデータを保存するテーブルです。

```
CREATE TABLE t_userdir
(
    uid          BIGINT,
    username     VARCHAR(100),
    longname     VARCHAR(200),
    password     VARCHAR(100),
    email       VARCHAR(100),
    description  VARCHAR(200),
    gid         BIGINT,
    timeexpire  BIGINT,
    timemake    BIGINT
);
```

```
CREATE UNIQUE INDEX idx_userdir_uid ON t_userdir( uid );
CREATE INDEX idx_userdir_username ON t_userdir( username );
CREATE INDEX idx_userdir_timeexpire ON t_userdir( timeexpire );
```

4. **[Configuration] > [Database]** ページで以降の項目を設定してください。

Registered Database			
	項目名	デフォルト値	説明
	On/Off	Off	レジスタ・データベース用にサード・パーティーのデータベースの使用を有効 (on) ・無効 (off)にする。
	Registered Database URL		Brekeke SIP Server が参照するサード・パーティーのデータベースの URL。(例: jdbc:mysql://localhost/mydb)
	Registered Database Driver		使用する JDBC ドライバ名 (ex. com.mysql.jdbc.Driver)
	User Name		データベースにアクセスするためのユーザ名
	Password		データベースにアクセスするためのパスワード

Users Database			
	Item	Default value	Explanation
	On/Off	Off	ユーザ・データベース用にサード・パーティーのデータベースの使用を有効 (on) ・無効 (off)にする。
	Registered Database URL		Brekeke SIP Server が参照するサード・パーティーのデータベースの URL。 (例: jdbc:mysql://localhost/mydb)
	Registered Database Driver		使用する JDBC ドライバ名 (ex. com.mysql.jdbc.Driver)
	User Name		データベースにアクセスするためのユーザ名
	Password		データベースにアクセスするためのパスワード

5.8.6. Login Password

Brekeke SIP Server 管理ツールのパスワードを設定します。

SIP リクエストの認証で使用するパスワードの設定は「5.4 User Authentication ページ」を参照し、**[User Authentication]**ページで行ってください。

デフォルトの管理者用アカウントは User ID: sa Password: sa です。

項目名	デフォルト値	説明
Type a new password	sa	Brekeke SIP Server 管理ツールにアクセスするためのパスワードです。

ボタン	説明
Apply	パスワードの変更を有効にします。

5.8.7. Upgrade Software

Brekeke SIP Server のアップグレードを行います。

- 1) あらかじめ**[Start/Shutdown]**ページでサーバをシャットダウンします。
- 2) **[Browse]** (または**[参照]**) ボタンでファイルを指定し、**[Upload]** ボタンを押します。

5.8.8. Update Product ID

Brekeke SIP Server 用のプロダクト ID をアップデートします。

- 1) **[Restart/Shutdown]** ページから Brekeke SIP Server をシャットダウンします。
- 2) **[Update Product ID]** で新たなプロダクト ID を入力し、**[I agree]** ボタンをクリックします。

6. ダイアル・プラン

6.1. ダイアル・プランとは？

Brekeke SIP Server のダイアル・プランでは、ルーティング先を決定するためのルールを定義します。また、任意のルールによるフィルタリングやサーバの環境設定、特定の SIP ヘッダを書き換えることによる情報の漏洩防止なども可能です。これらのルールは正規表現で記述されるため、柔軟に定義できます。

以下はダイアル・プランで実現できる主な機能です。

- ◆ ルーティング先の決定
- ◆ 認証の要求
- ◆ フィルタリング
- ◆ SIP ヘッダの追加・削除・書き換え
- ◆ 複数サーバへの負荷の分散
- ◆ サーバ環境変数の設定
- ◆ RTP パケットのリレー有無
- ◆ セッション・プラグインのロード

ダイアル・プランの応用については、「Brekeke SIP Server チュートリアル - ダイアル・プラン編」も合わせて参照してください。

6.2. 編集

ダイアル・プランの編集は管理ツールの[Dial Plan]ページで行います。編集したいルールの場所で



をクリックすることによって、編集できます。詳細は、「5.3 Dial Plan ページ」を参照してください。

また、ダイアル・プラン・ファイルを直接編集することもできます。この場合、ダイアル・プラン・ファイルを編集した後に Brekeke SIP Server を再起動してください。

ダイアル・プラン・ファイルの場所は、

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

6.3. 文法

ダイヤル・プランでは Matching Patterns に条件を定義し、その条件が満足した場合の処理内容を Deploy Patterns に定義します。

6.3.1. Matching Patterns

ルールを適用するための条件を定義します。条件には SIP ヘッダやパケットの送信元アドレス、ポート番号、時間などを指定できます。

条件の定義は SIP ヘッダ・フィールド名、条件関数名または環境変数名と、その値を照合するパターン文字列のペアです。複数のペアを定義することによって、条件を絞り込むことができます。

パターン文字列は正規表現で記述されます。このとき、括弧 '(') 内の文字列は以降の Matching Patterns や Deploy Patterns で参照することができます。

Matching Patterns の構文

SIP ヘッダ・フィールド名 = パターン文字列

&環境変数名 = パターン文字列

\$条件関数名 = パターン文字列

\$条件関数名(引数) = パターン文字列

パターン文字列で使用可能な主な正規表現は以下の通りです。

記号	意味
!	! が文字列パターンの前におかれると、NOT(否定) を意味します。
^	行の先頭
\$	行の終わり
[abc]	[] 内にある a,b,c いずれか任意の 1 文字
[^ abc]	[] 内の a,b,c いずれでもない任意の 1 文字
.	改行以外の任意の 1 文字
X+	前の文字(ここでは X)が 1 回以上
X*	前の文字(ここでは X)が 0 回以上
X{n}	前の文字(ここでは X)が n 回
X{n, }	前の文字(ここでは X)が n 回以上
X{n, m}	前の文字(ここでは X)が n 回以上 m 回以下
(括弧内の文字)	() カッコ内の文字列がバッファに保存され、%n で参照できます。

管理ツールの Dial Plan 編集ページにおいて、条件を **[Matching Patterns]**フィールドに追加する場合は、[...]ボタン(**[Variable]**と**[Value]**フィールドの間にあります。) を押し、**[Variable]**フィールドに表示されるドロップダウン・メニューから変数名を選択するか、**[Variable]**フィールドに変数名を入力してください。そして、**[Value]**フィールドに評価したいパターン文字列を入力し、最後に**[insert]**ボタンを押してください。詳細は「5.5.2 Dial Plan 編集ページ」を参照してください。

1) SIP ヘッダ・フィールド名

SIP ヘッダを条件としてルールを適用したい場合は、SIP ヘッダ・フィールド名とパターン文字列のペアを指定します。

構文:

SIP ヘッダ・フィールド名 = パターン文字列

例:

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

From:で示されたSIP URI が “sip:user@domain.com” のとき。

To = sip:11@

To:で示されたユーザー名が “11” のとき。

To = sip:9(.+)@

To:で示されたユーザー名のプリフィックスが “9” のとき。

To = sip:(....)@

To:で示されたユーザー名の長さが4文字のとき。

Supported = timer

Supported: が”timer”を含むとき。

Expires = ^[0-5]\$

Expires: の値が 0 以上 5 以下のとき。

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

Contact:のユーザー名がアルファベットのとき。

2) 環境変数

左辺のプリフィックスに'&'が付く場合は環境変数として処理されます。変数名は大文字・小文字を区別しません。

構文:

&環境変数名 = パターン文字列

例:

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

サーバ名(環境変数 sv.name)の値が "main-sv" のとき。

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

リングング・タイムアウト(環境変数 net.sip.timeout.ringing)の値が 5000~5999 のとき。

3) 条件関数

左辺のプリフィックスに'\$'が付く場合は条件関数として処理されます。関数名は大文字・小文字を区別しません。一部の条件関数は引数を持つことができます。

構文:

\$条件関数名 = パターン文字列

\$条件関数名(引数) = パターン文字列

関数呼び出しのフォーマット:

関数名(SIP ヘッダ・フィールド名)

引数として SIP ヘッダ・フィールド名を指定することによって、該当する SIP ヘッダの値が関数に渡されます。

構文例: \$func(From)

From:ヘッダの値が関数 func に渡されます。

関数名(&環境変数名)

引数として環境変数名を指定することによって、該当する変数の値が関数に渡されます。環境変数名にはプリフィックスとして‘&’が付きます。

- 環境変数は Deploy Patterns またはプロパティ・ファイルで設定できます。

構文例: `$func(&net.sip.timeout.ringing)`

環境変数 `net.sip.timeout.ringing` の値が関数 `func` に渡されます。

関数名(\$条件関数名)

引数として条件関数名を指定することによって、その関数の戻り値が呼び出し元の関数に渡されます。条件関数名にはプリフィックスとして‘\$’が付きます。

また、引数の関数名が呼び出し元の関数名と同じである場合は、再帰的に関数が呼び出されません。

構文例: `$func1($func2)`

関数 `func2` の戻り値が関数 `func1` に渡されます。

構文例: `$func1($func2($func3))`

関数 `func3` の戻り値が関数 `func2` に渡され、関数 `func2` の戻り値が関数 `func1` に渡されます。

構文例: `$func($func(To))`

`To`:ヘッダの値が関数 `func` に渡され、その戻り値が再び関数 `func` に渡されます。

関数名(“文字列”)

引数として文字列を指定することによって、その文字列が関数に渡されます。文字列はダブルクォテーション(“”)で囲まれている必要があります。

構文例: `$func(“string”)`

文字列 “string” が関数 `func` に渡されます。

\$addr**意味:**

送信元 IP アドレス

構文:

\$addr

説明:

パケットの送信元 IP アドレスを返します。

例:

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

パケット送信元 IP アドレスがループバック・アドレス(127.0.0.1)のとき。

\$addr = ^192\.168\.

パケット送信元 IP アドレスのプリフィックスが "192.168." のとき。

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

パケット送信元 IP アドレスが 172.16.0.1 から 172.16.0.5 の範囲内にあるとき。

\$date**意味:**

年月日

構文:

\$date

\$date(format)

format - 時間のフォーマット

説明:

現在の年月日の文字列を返します。

文字列のフォーマットは引数 format で指定します。デフォルトのフォーマットは "yyyy/MM/dd" です。

フォーマットは以下の文字から構成されます。

文字	意味	文字	意味
y	年	m	分
M	月	s	秒
d	日	S	ミリ秒
H	時		

例:

```
$date = 2010/06/03
```

年月日が 2010 年 6 月 3 日のとき。

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

日付の下一桁が1または5のとき。

つまり各月の 1 日,5 日,11 日,15 日,21 日,25 日,31 日のとき。

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

”月-日-年”のフォーマットで文字列を取得し、2010 年 6 月 3 日であるかどうか比較します。

\$geturi**意味:**

URI 文字列の取得

構文:

```
$geturi( str )
```

str - 文字列

説明:

文字列 str から SIP URI を取得します。

例:

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

From:ヘッダから SIP URI 部分を取得し、”sip:user@domain.com”であるかどうかを比較します。

以下の条件と同じ意味になります。

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

```
$geturi( &sv.admin.sip ) = sip:admin@host$
```

環境変数 sv.admin.sip の値から SIP URI 部分を取得し、”sip:admin@host”であるかどうかを比較します。

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

request-line(条件関数 request の戻り値)から SIP URI 部分を取得し、

”sip:1234@192.168.0.1”であるかどうかを比較します。

\$localhost

意味:

ローカルホスト有無

構文:

`$localhost`

`$localhost(str)`

str - SIP URI または IP アドレス または FQDN

説明:

引数で与えられた SIP URI またはアドレスがローカルホストを示しているかどうかを判断します。

ローカルホストの場合は “true”、それ以外は “false” を返します。

引数が省略された場合は、パケットの送信元 IP アドレスがローカルホストであるかどうかを判断します。

- 環境設定でネットワーク・インターフェースとして指定されているアドレスもローカルホストとして判断されます。

例:

```
$localhost = true
```

パケットの送信元がローカルホストのとき。

```
$localhost( $addr ) = true
```

パケットの送信元がローカルホストのとき。(引数を省略した場合と同じです。)

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

From ヘッダの SIP URI がローカルホストを示していないとき。

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

192.168.0.1 がローカルホストのとき。

\$outbound

意味:

アウントバンド有無

構文:

`$outbound`

`$outbound(str)`

str - SIP URI または IP アドレス または FQDN

説明:

引数で与えられた SIP URI またはアドレスがアウトバンド(自サーバ以外のアドレスまたはポート番号)のアドレスを示しているかどうかを判断します。

アウトバンドの場合は “true”、それ以外は “false” を返します。

引数が省略された場合は、リクエスト URI がアウトバンドを示しているかどうかを判断します。

たとえば、自サーバが 192.168.0.1:5060 で稼働しているとき、192.168.0.2 や 192.168.0.1:6060 はアウトバンドとして扱われます。

例:

```
$outbound = true
```

リクエスト URI がアウトバンドのアドレスを示しているとき。

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

リクエスト URI がアウトバンドのアドレスを示しているとき。(引数を省略した場合と同じです。)

```
$out bound( To ) = false
```

To ヘッダの SIP URI がアウトバンドを示していないとき。

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

host がアウトバンドのアドレスのとき。

\$port**意味:**

送信元ポート番号

構文:

```
$port
```

説明:

パケットの送信元ポート番号を返します。

例:

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

パケット送信元ポート番号が 5060 のとき。

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

パケット送信元ポート番号が 5000 から 5099 の範囲内にあるとき。

\$registered

意味:

レジスト済みの有無

構文:

`$registered`

`$registered(str)`

str - SIP URI または ユーザー名

説明:

引数で与えられた SIP URI または ユーザー名がレジスタ・データベースに登録されているかどうかを判断します。

該当するユーザーが登録されている場合は “true”、それ以外は “false” を返します。

引数が省略された場合は、リクエスト URI が示すユーザーが登録済みであるかどうかを判断します。

例:

```
$registered = true
```

リクエスト URI のユーザーがレジスト済みのとき。

```
$registered( $request ) = true
```

リクエスト URI のユーザーがレジスト済みのとき。(引数を省略した場合と同じです。)

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

発信者(From:ヘッダのユーザー)がレジスト済みのとき。

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

ユーザー “user” がレジスト済みでないとき。

\$registeredaddr

意味:

レジストされているコンタクト先の IP アドレス

構文:

`$registeredaddr`

`$registeredaddr (str)`

str - SIP URI または ユーザー名

説明:

引数で与えられた SIP URI またはユーザー名について、レジスタ・データベースに登録されているコンタクト先の IP アドレスを返します。

引数が省略された場合は、リクエスト URI が示すユーザーについて、登録されている IP アドレスを返します。

該当するユーザーが存在しない場合、条件は成立しません。

例:

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

リクエスト URI のユーザーが 192.168.0.1 から登録されているとき。

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

発信者(From ヘッダのユーザー)が 192.168.0.200 から登録されているとき。

```
$registeredaddr( "user" ) = ^192\.168\.0\.
```

ユーザー"user"が 192.168.0.x から登録されているとき。

\$registereduri**意味:**

レジスタされているコンタクト先の SIP URI

構文:

```
$registereduri
```

```
$registereduri ( str )
```

str - SIP URI または ユーザー名

説明:

引数で与えられた SIP URI またはユーザー名について、レジスタ・データベースに登録されているコンタクト先の SIP URI を返します。

引数が省略された場合は、リクエスト URI が示すユーザーについて、登録されている SIP URI を返します。

該当するユーザーが存在しない場合、条件は成立しません。

例:

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

リクエスト URI のユーザーのコンタクト先 SIP URI が 100@host のとき。

```
$registereduri ( "user" ) = sip:admin@
```

ユーザー"user"のコンタクト先 SIP URI のユーザー名が admin のとき。

\$request

意味:

SIP リクエスト・ライン

構文:

\$request

説明:

パケットのリクエスト・ラインを返します。

例:

```
$request = sip:100@host[>]*
```

リクエスト URI が “sip:100@host” のとき。

```
$request = ^INVITE
```

リクエストが INVITE のとき。

\$sid

意味:

セッション ID

構文:

\$sid

説明:

セッション ID を返します。

セッション ID はセッションごとに割り当てられるユニークな番号です。

例:

```
$sid = ^100$
```

セッション ID が 100 のとき。

```
$sid = [02468]$
```

セッション ID が偶数のとき。

\$time

意味:

現在の時刻

構文:

\$time

\$time (format)

format - 時間のフォーマット

説明:

現在時刻の文字列を返します。

文字列のフォーマットは引数 `format` で指定します。デフォルトのフォーマットは "HH:mm:ss" です。フォーマットの詳細は `$date` 関数を参照してください。

例:

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

時刻が 9 時 26 分 40 秒のとき。

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

時刻が午前 0 時から午前 9 時のあいだのとき。

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

ミリ秒の単位が偶数のとき。

6.3.2. Deploy Patterns

条件が成立したときの処理内容を定義します。処理内容には SIP ヘッダやルーティング先アドレス、環境変数、ロードするプラグイン名などを指定できます。

処理内容の定義は SIP ヘッダ・フィールド名、処理変数名または環境変数名と、そこに設定する値のペアです。1つのルールに複数の処理を定義することもできます。

設定値では Matching Patterns でマッチした文字列を参照することができます。設定値の中で '%n'(n は数値)と記述されていた場合、Matching Pattern のパターン文字列内の n 番目の括弧 '(' で囲まれた文字列が参照されます。

Deploy Patterns の構文

SIP ヘッダ・フィールド名 = 設定値

&環境変数名 = 設定値

\$処理変数名 = 設定値

管理ツールの Dial Plan 編集ページにおいて、処理内容を[**Deploy Patterns**]フィールドに追加する場合は、[...]ボタン([**Variable**]と[**Value**]フィールドの間にあります。)を押し、[**Variable**]フィールドにあらわれるドロップダウン・メニューから変数名を選択するか、[**Variable**]フィールドに変数名を入力してください。そして、[**Value**]フィールドに設定したい値を入力し、最後に[**insert**]ボタンを押してください。詳細は「5.5.2 Dial Plan 編集ページ」を参照してください。

1) SIP ヘッダ・フィールド名

処理内容に SIP ヘッダ・フィールド名を指定することによって、SIP ヘッダの置換、追加、または削除をすることができます。

指定された SIP ヘッダ・フィールド名がすでに SIP パケット内に存在すれば、新しい設定値で置換されます。SIP パケットに存在しなければ新規に追加されます。

また、設定値が空文(長さ0の文字列)の場合、その SIP ヘッダはパケットから削除されます。

SIP ヘッダ・フィールド名が To の場合、その設定値に応じてルーティング先が以下のように決定されます。

To = sip:username@host の場合
host で指定されたアドレスにセッションをルーティングします。

To = sip:username@ の場合
レジスタ・データベースに登録されているユーザ名 "user" のコンタクト先アドレスにセッションをルーティングします。

構文:

SIP ヘッダ・フィールド名 = 設定値

例:

From = sip:admin@192.168.0.1

From: を "sip:admin@192.168.0.1" に置換します。

To = sip:boss@192.168.0.100

To: を "sip:boss@192.168.0.100" に置換します。これによりセッションは置換された新しい SIP URI にルーティングされます。

To = sip:sales@

セッションを ユーザ "sales" のコンタクト先アドレスにルーティングします。

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

From: を sip:1650111@domain に置換します。また発信者のディスプレイ・ネームを Ted にします。

Expires = 300

Expires: の値を 300 にします。

User-Agent =

User-Agent:ヘッダを削除します。

Refer-To = sip:user@server

Refer-To:を user@server に置換します。

2) 環境変数

左辺のプリフィックスに'&'が付く場合は環境変数として処理されます。変数名は大文字・小文字を区別しません。

ここで設定される環境変数は、ルールが適用されたセッション内のみで有効です。システム全体の環境設定はプロパティファイル または **[Configuration]** ページでおこなってください。

構文:

&環境変数名 = 設定値

例:

```
&net.sip.timeout.ringing = 10000
```

リングング・タイムアウト(環境変数 `net.sip.timeout.ringing`)の値を10000に設定。

```
&net.sip.addrecordroute = false
```

Record-Route:ヘッダを付加しない。

(環境変数 `net.sip.addrecordroute` の値を `false` に設定。)

```
&net.rtp.audio.payloadtype = 0
```

SDP 内の audio のペイロード・タイプを PCMU に置換する。

(環境変数 `net.rtp.audio.payloadtype` の値を `0` に設定。)

3) 処理変数

左辺のプリフィックスに'\$'が付く場合は処理変数として処理されます。変数名は大文字・小文字を区別しません。

構文:

\$処理変数名 = 設定値

\$action

意味:

レスポンス送信

構文:

`$action = SIP レスポンス番号`

説明:

リクエストに対する SIP レスポンスの番号を設定します。

レスポンスが返されたリクエストの packets はルーティングされません。

例:

```
$action = 200
```

レスポンス 200 OK を返します。

```
$action = 603
```

レスポンス 603 Decline を返します。

\$auth

意味:

認証の有無

構文:

`$auth = true または false`

説明:

リクエストを認証するかどうかを設定します。

“true”の場合は有効。“false”の場合は無効。デフォルト値は環境設定された値です。

例:

```
$auth = true
```

リクエストを認証します。

\$continue

意味:

ルール評価継続の有無

構文:

`$continue = true または false`

説明:

複数のルールを処理させるための変数です。

“true”の場合は、ダイヤル・プランの次のルールを継続して評価します。

“false”の場合は、評価を継続しません。デフォルトは“false”です。

つまり、Matching Patterns の条件が満たされ、かつ Deploy Patterns が \$continue=true 含む限り、ルールの評価が継続されます。

例:

```
$continue = true
```

次のルールを継続して評価します。

\$ifdst**意味:**

送信先方向のインターフェース・アドレス

構文:

```
$ifdst = IP アドレス または FQDN
```

説明:

セッションの送信先(着側)方向のサーバのインターフェース・アドレスを設定します。

このアドレスは、Via:、Record-Route: などに使用されます。

例:

```
$ifdst = 192.168.0.100
```

送信先方向のインターフェース・アドレスとして 192.168.0.100 を設定します。

\$ifsrc**意味:**

送信元方向のインターフェース・アドレス

構文:

```
$ifsrc = IP アドレス または FQDN
```

説明:

セッションの送信元(発側)方向のサーバのインターフェース・アドレスを設定します。

このアドレスは、Via:、Record-Route: などに使用されます。

例:

```
$ifsrc = 192.168.1.200
```

送信元方向のインターフェース・アドレスとして 192.168.1.200 を設定します。

\$nat

意味:

NATトラバーサル処理の有無

構文:

`$nat = true` または `false`

説明:

NATトラバーサル処理をするかどうかを設定します。

“true”の場合は有効。“false”の場合は無効。“auto”の場合は自動的に判断します。

デフォルトは“auto”です。

NATトラバーサル処理を有効にすると、RTP リレー(変数\$rtplib)も合わせて有効になります。

また、NATトラバーサル処理を無効にすると、サーバが自動的に NATトラバーサルの必要性を検出しても、NATに関する処理は行われません。

例:

```
$nat = true
```

NATトラバーサル処理を有効にします。

\$replaceuri

意味:

From 及び To ヘッダー置換の有無

構文:

`$replaceuri = true` または `false`

説明:

From 及び To ヘッダーを置換するかどうかを設定します。

“true”の場合は有効。“false”の場合は無効。“auto”の場合は自動的に判断します(NATトラバーサル有無など)。

デフォルトは“auto”です。

置換が有効である場合、たとえば NAT 外に送られるパケットに NAT 内のアドレスを含めないなどの処理が行われます。

例:

```
$replaceuri = false
```

From 及び To ヘッダーの置換をしません。

\$rtp

意味:

RTP リレーの有無

構文:

`$rtp = true` または `false`

説明:

RTP パケットをサーバでリレーするかどうかを設定します。

“true”の場合は有効。“false”の場合は無効。“auto”の場合は自動的に判断します(NAT トラバースの有無など)。デフォルトは環境設定された値です。

例:

```
$rtp = true
```

RTP リレーを有効にします。

\$session

意味:

セッション・プラグインのロード

構文:

`$session = セッション・プラグイン名`

説明:

利用するセッション・プラグイン名を設定します。

例:

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

プラグインとして `com.sample.radius.proxy.RadiusAcct` クラスを設定します。

\$target

意味:

ルーティング先アドレス

構文:

`$target = IP アドレス` または `FQDN`

説明:

セッションのルーティング先を設定します。

例:

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

セッションを `provider.domain` にルーティングします。

7. Upper Registration

Upper Registration は、クライアントから送られた REGISTER リクエストを Brekeke SIP Server を経由して、指定された他のサーバ(以下、上位サーバ)に転送する機能です。これによってクライアントは上位サーバにも登録されることになり、上位サーバを利用している他のユーザーからも着信を受けることが可能になります。

Upper Registration は Brekeke SIP Server 側で REGISTER リクエストの転送先サーバを指定するため、クライアント側は通常の設定で構いません。

- ✓ Thru Registration は、クライアント側で転送先サーバを指定する必要があります。

1) Brekeke SIP Server での設定

「5.8.3 SIP(Advanced)」を参照し、[Configuration]ページの [SIP(Advanced)] で Upper Registration を設定してください。

項目名	設定値	説明
On/Off	on	Upper Registration を有効にします。
Register Server (IP or FQDN)	上位サーバのアドレス	Upper Registration 先のレジスタサーバの IP アドレスまたは FQDN を指定します。 たとえば、ITSP 側サーバのアドレスなどを指定します。

2) クライアントでの設定

各クライアントで以下のように設定してください。設定方法は各クライアントのドキュメントを参照してください。

項目名	設定値
SIP プロキシサーバ	Brekeke SIP Server のアドレス
レジストラ	Brekeke SIP Server のアドレス
アウトバンド・プロキシ	Brekeke SIP Server のアドレス
ユーザー名	任意のユーザー名 但し、上位サーバでリクエストの認証がされている場合は、上位サーバに登録されているユーザー名を指定してください。
パスワード	上位サーバでリクエストの認証がされている場合は、上位サーバに登録されているパスワードを指定してください。

8. Thru Registration

Thru Registration は、クライアントから送られた REGISTER リクエストのリクエスト URI が他のサーバのアドレスを示しているとき、そのリクエストを Brekeke SIP Server を経由して、該当する他のサーバ(以下、上位サーバ) に転送する機能です。これによってクライアントは上位サーバにも登録されることになり、上位サーバを利用している他のユーザーからも着信を受けることが可能になります。Thru Registration はクライアントごとに設定が必要であるため、それぞれが別々の上位サーバを指定することができます。

- ✓ Upper Registration は Brekeke SIP Server で 1 つの上位サーバしか指定できません。

1) Brekeke SIP Server での設定

「5.8.3 SIP(Advanced)」を参照し、**[Configuration]** ページの **[SIP(Advanced)]** で Thru Registration を設定してください。

項目名	設定値	説明
On/Off	on	Thru Registration を有効にします。

2) クライアントでの設定

各クライアントで以下のように設定してください。設定方法は各クライアントのドキュメントを参照してください。

項目名	設定値
SIP プロキシ・サーバ	Brekeke SIP Server のアドレス 但し、アウトバンド・プロキシを指定可能なクライアントは、ここに上位サーバのアドレスを指定してください。
レジストラ	上位サーバのアドレス
アウトバンド・プロキシ	Brekeke SIP Server のアドレス
ユーザー名	任意のユーザー名 但し、上位サーバでリクエストの認証がされている場合は、上位サーバに登録されているユーザー名を指定してください。
パスワード	上位サーバでリクエストの認証がされている場合は、上位サーバに登録されているパスワードを指定してください。

9. NATトラバースル

この節では Brekeke SIP Server が持つ NATトラバースル機能の設定方法を説明します。

9.1. Near-End NATトラバースル

Brekeke SIP Server とクライアントが同じファイアーウォール内に存在するとき、ファイアーウォールの外側と通信をするために Near-End NATトラバースルが使われます。

必要な設定は以下の通りです。

1) Brekeke SIP Server でのインターフェース設定

ルータのグローバル・サイドの IP アドレスまたは FQDN を Brekeke SIP Server のネットワーク・インターフェースとして設定します。

設定場所は、管理ツールの[Configuration]ページ → [System] → [Interface Address]です。

2) ルータまたはファイアーウォールでのポート・フォワーディング設定

ルータのポート・フォワーディング設定で Brekeke SIP Server が使用するポートを Brekeke SIP Server にフォワードするように指定します。

また、ファイアーウォールでパケット・フィルタリングがされている場合は、Brekeke SIP Server が使用するポートを開けてください。

下記は Brekeke SIP Server が使用するポートです。

プロトコル	ポート番号 (デフォルト)	設定場所
SIP (UDP)	5060	[Configuration] > [SIP(General)]で変更できます。
RTP (UDP)	10000-10999	[Configuration] > [RTP]で変更できます。

9.2. Far-End NATトラバーサル

クライアントがインターネットを介した遠隔のファイアーウォール内に存在する場合、Brekeke SIP Server を利用して他のクライアントと通信するために Far-End NATトラバーサルが使われます。必要な設定は以下の通りです。また、Brekeke SIP Server もファイアーウォール内に存在する場合は、Near-End NAT の設定も合わせて行ってください。

1) Brekeke SIP Server でのキープ・アライブ設定

Far-End NAT ではサーバとクライアント間のパケット交換のためにクライアント側のルータにおいてポート・マッピングを維持する必要があります。ポート・マッピングがクリアされると、サーバから送られたパケットがクライアントに到達できない可能性があります。

Brekeke SIP Server はポート・マッピングの維持のためにキープ・アライブ機能によって一定間隔ごとにダミーのパケットをクライアントに送出します。ダミー・パケットの送出間隔が長いとポート・マッピングがクリアされますので適宜に調整してください。

また、ルータの種類によってはキープ・アライブが有効にならない場合もあります。その場合は次項を参考にしてルータでポート・フォワーディングを設定してください。

キープ・アライブの設定場所は、管理ツールの [Configuration] > [SIP(General)] > [NAT traversal]です。

項目名	設定値	説明
Keep address/port mapping	on	キープ・アライブを有効にします。
Interval (ms)	ネットワーク環境に依存	ダミー・パケットを送出する間隔です。単位はミリ秒。デフォルトは 120000 ミリ秒(=2 分)です。ポート・マッピングがクリアされない程度に短い時間を設定してください。

2) クライアント側のルータまたはファイアーウォールでのポート・フォワーディング設定

Brekeke SIP Server によるキープ・アライブを使用してもパケットがクライアントに到達しない場合は、クライアント側のルータでポート・フォワーディングを設定する必要があります。

ポート・フォワーディングではクライアントが使用するポートをクライアントにフォワードしてください。また、ファイアーウォールでパケット・フィルタリングがされている場合は、クライアントが使用するポートを開けてください。

クライアントが使用するポート番号については、クライアントの設定画面またはドキュメントを参照してください。

10. クライアントの設定

Brekeke SIP Server を利用して通信を行うためには、クライアントを用意し適切に設定をする必要があります。

10.1. クライアントの用意

SIP プロトコルをサポートした、ソフトフォン、IP 電話機、VoIP ゲートウェイ、アナログ・テレフォン・アダプタ(ATA)、IM クライアントなどを用意してください。

ソフトウェア・タイプのクライアントの中には、無料で使用・試用できる製品もあります。

10.2. 設定項目

以下は、基本的なクライアントの設定項目とその設定内容です。

ここに挙げた設定項目が存在しないクライアントもあります。

項目名	設定内容
SIP プロキシ・サーバ	Brekeke SIP Server のアドレスを設定してください。 サーバのポート番号を指定する必要がある場合は 5060 を設定してください。 サーバのポート番号を変更したい場合は「5.8.2. SIP(General)」を参照してください。
レジストラ	Brekeke SIP Server のアドレスを設定してください。
アウトバンド・プロキシ	Brekeke SIP Server のアドレスを設定してください。
ドメイン	Brekeke SIP Server のアドレスを設定してください。
Realm	Brekeke SIP Server のアドレスを設定してください。 但し、サーバがリクエスト認証をする場合は、サーバに登録されている Realm を設定してください。
ユーザー名	任意にユーザー名を設定してください。
認証ユーザー名	任意にユーザー名を設定してください。 但し、サーバがリクエスト認証をする場合は、サーバに登録されているユーザー名を設定してください。
パスワード	サーバがリクエスト認証をする場合は、サーバに登録されているパスワードを設定してください。
STUN	クライアントが NAT 内の場合、STUN を使うことによって NAT トラバースルがうまくいく場合があります。 逆に STUN を使うことによって、NAT トラバースルが失敗する場合があります。その場合は STUN の使用を無効にしてください。

10.3. レジスタの確認

クライアントの設定が適切にされている場合、そのクライアントのレジスタ情報を Brekeke SIP Server 管理ツールの[Registered List]ページで確認することができます。詳細は「5.3. Registered List ページ」を参照してください。

Appendix . A 用語解説

◆ 管理ツール (Admintool) または 管理画面 または アドミン・ツール

Brekeke SIP Server を管理するためのフロント・エンド環境です。

ウェブ画面であるため、ブラウザを使うことによってローカルやリモートからアクセスすることができます。管理ツールではサーバの起動・停止、状況の表示、環境の設定などができます。

➤ 参照: 「5. 管理ツール」

◆ クライアント (Client)

ユーザーが通信を開始したり受けたりするソフトウェアまたはハードウェアです。クライアントは SIP プロトコルをサポートしている必要があります。

たとえば、ソフトフォン、IM クライアント、IP 電話機などがクライアントに該当します。Brekeke SIP Server はこれらのクライアントを結びつける役割をします。

➤ 参照: 「10. クライアントの設定」

➤ 関連用語: サーバ、SIP、UA

◆ デプロイ・パターン (Deploy Patterns) または 処理パターン

ダイヤル・プランのルールで処理内容を定義するパターンです。処理内容として SIP ヘッダの置換や、SIP パケットの送信先の指定などを定義できます。

➤ 参照: 「5.5.2 Dial Plan 編集ページ」、「6.3.2. Deploy Patterns」

➤ 関連用語: ダイヤル・プラン、ルール、マッチング・パターン

◆ ダイヤル・プラン (Dial Plan)

Brekeke SIP Server が持つルーティング先を決定する方法のひとつです。ダイヤル・プランは複数のルールで構成され、各ルールは Matching Patterns と Deploy Patterns のペアです。Matching Patterns の条件に一致した場合のみ、その対の Deploy Patterns が処理されます。管理ツールの[Dial Plan]ページで閲覧と編集ができます。詳細は「6. ダイヤルプラン」を参照してください。

➤ 参照: 「1.1 Brekeke SIP Server とは?」、「5.5 Dial Plan ページ」、「6 ダイヤル・プラン」

➤ 関連用語: ルール、デプロイ・パターン、マッチング・パターン

◆ 環境変数 (Environment Variable)

Brekeke SIP Server の動作や管理情報、各種内部パラメータを指定するための変数です。プロパティファイルで指定することができます。また、一部の環境変数は[**Configuration**]ページの設定項目に対応しています。

セッションごとに環境変数を指定したい場合は、ダイヤル・プランの Deploy Patterns に記述します。

- 参照: 「5.8 Configuration ページ」、「6 ダイアル・プラン」
- 関連用語: デプロイ・パターン

◆ Far-End NATトラバーサル

クライアントが遠隔のファイアーウォール内に存在する場合の NATトラバーサルです。

- 参照: 「1.1 Brekeke SIP Server とは?」、「9 NATトラバーサル」
- 関連用語: NATトラバーサル、Near-End NATトラバーサル

◆ ITSP

Internet Telephony Service Provider の略。インターネット電話サービスを提供する通信事業者です。IP 電話事業者とも呼ばれます。

◆ マッチング・パターン (Matching Patterns) または 条件パターン

ダイヤル・プランのルールで条件を定義するパターンです。条件には SIP ヘッダやパケットの送信元アドレスなどを正規表現で指定できます。

- 参照: 「5.5.2 Dial Plan 編集ページ」、「6.3.1. Matching Patterns」
- 関連用語: ダイアル・プラン、ルール、デプロイ・パターン

◆ NATトラバーサル

セッションに参加する各クライアントがそれぞれ異なるファイアーウォール内に存在する場合、NATトラバーサル機能によって両者を結びつけます。また、このとき環境に応じて RTP リレーをします。Brekeke SIP Server の NATトラバーサル機能は、Far-End NAT と Near-End NAT の両方に対応しています。

- 参照: 「1.1 Brekeke SIP Server とは?」、「9 NATトラバーサル」
- 関連用語: Near-End NATトラバーサル、Far-End NATトラバーサル、RTP リレー

◆ Near-End NATトラバーサル

サーバとクライアントが同じファイアーウォール内に存在する場合の NATトラバーサルです。

- 参照: 「1.1 Brekeke SIP Server とは?」、「9 NATトラバーサル」
- 関連用語: NATトラバーサル、Far-End NATトラバーサル

◆ レジスタ・データベース (register database)

クライアントから送られた REGISTER リクエストをもとに、そのクライアントのアドレスを記録するデータベースです。Brekeke SIP Server は必要に応じてデータベースに記録されたアドレスをルーティング先として利用します。管理ツールの[Registered List]ページで閲覧ができます。

- 参照: 「1.1 Brekeke SIP Server とは?」、「5.3 Registered List ページ」
- 関連用語: Thru レジストレーション、Upper レジストレーション

◆ RTP

Real-time Transport Protocol の略。クライアントがメディア(音声、ビデオなど) の送受信に用いるプロトコルです。詳細は RFC1889,1890 などを参照してください。

- 参照: 「5.8.4 RTP」
- 関連用語: SIP、RTP リレー

◆ RTP リレー または RTP トンネル または RTP 中継

通常、RTP パケットはクライアント間で直接交換されますが、その実現が難しい環境においては、Brekeke SIP Server が必要に応じて RTP パケットをリレー(中継) します。このときポート番号 10000-10999(デフォルト) の範囲を使用します。

Brekeke SIP Server は音声以外にもビデオなどのメディア・ストリームの RTP パケットをリレーすることができます。

- 参照: 「1.1 Brekeke SIP Server とは?」、「5.8.4 RTP」
- 関連用語: NAT トラバーサル、RTP

◆ ルール (rule) または ダイアル・プラン・ルール

ダイアル・プランを構成する Matching Patterns と Deploy Patterns のペアです。

- 参照: 「5.5.2 Dial Plan 編集ページ」、「6.3 文法」
- 関連用語: ダイアル・プラン、デプロイ・パターン、マッチング・パターン

◆ セッション (session)

INVITE リクエストによって開始されます。音声通信の場合、通常は 1 通話につき 1 セッションです。セッションは BYE リクエストや初回の INVITE リクエストに対するエラーレスポンスが処理されるまで継続します。セッションは管理ツールの[Session List]ページで確認できます。

- 参照: 「5.4 Session List ページ」
- 関連用語: SIP

◆ **セッション ID (Session ID) または SID**

セッションごとに割り当てられる番号です。

- 参照: 「5.4 Session List ページ」
- 関連用語: セッション (session)

◆ **サーバ (Server)**

本ドキュメントでは、特に断りがないかぎり Brekeke SIP Server を意味します。

- 参照: 「1.1 Brekeke SIP Server とは?」

◆ **SIP**

Session Initiation Protocol の略。クライアントやサーバがセッションの生成などのコントロール (呼制御) に用いるプロトコルです。詳細は RFC3261 などを参照してください。

Brekeke SIP Server はクライアントから送られた SIP パケットを必要に応じて編集し、最適なルーティング先に送信します。ポート番号 5060(デフォルト) を使用します。

- 参照: 「1.1 Brekeke SIP Server とは?」
- 関連用語: RTP、セッション、サーバ、クライアント

◆ **Thru レジストレーション (Thru Registration)**

クライアントから送られた REGISTER リクエストのリクエスト URI が自サーバを示していないとき、その REGISTER リクエストをリクエスト URI で指定されたアドレスに転送する機能です。

- 参照: 「1.1 Brekeke SIP Server とは?」、「5.8.3 SIP(Advanced)」、「8. Thru Registration」
- 関連用語: レジスタ・データベース、Upper レジストレーション

◆ **UA または ユーザー・エージェント (user agent)**

- 関連用語: クライアント

◆ **ユーザー・ディレクトリ・データベース (user directory database)**

SIP リクエスト認証のためのユーザー情報としてユーザー名やパスワード等を保持するデータベースです。管理ツールの[Authentication]ページで閲覧・編集ができます。

Brekeke SIP Server でユーザーの認証を行う場合には、あらかじめユーザー・ディレクトリ・データベースに、該当のユーザーを登録しておく必要があります。

- 参照: 「5.6 Authentication ページ」「5.8.2 SIP(General)」

◆ Upper レジストレーション (Upper Registration)

クライアントから送られた REGISTER リクエストを、指定された他のサーバに転送する機能です。この機能によって、クライアントは 1 つの Brekeke SIP Server に REGISTER リクエストを送るだけで、他のサーバにも登録されることが可能です。

- 参照: 「1.1 Brekeke SIP Server とは?」、「5.8.3 SIP(Advanced)」、「7. Upper Registration」
- 関連用語: レジスタ・データベース、Thru レジストレーション