

OnDO SIP Server

Version 1.5

チュートリアル - ダイヤル・プラン編

Brekeke Software, Inc.

バージョン

OnDO SIP Server v1.5 チュートリアル - ダイアル・プラン編 (2005 年 12 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 ©2005 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、複写、複製、転載、多言語への翻訳、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	6
2.	ルーティング	6
2.1.	ルーティング先としてSIP URIを指定する場合	7
例 1	全ての通話を sip:user@host にルーティングする。.....	7
例 2	宛先のユーザー名がadminならば、sip:user@host にルーティングする。.....	7
例 3	宛先のSIP URIが sip:admin@serverならば、 sip:user@host にルーティングする。.....	7
例 4	宛先のユーザー名のプリフィックスが9ならば、同じユーザー名のままhostにルーティングする。.....	7
例 5	宛先のプリフィックスが9ならば、プリフィックスを削除したユーザー名でhostにルーティングする。.....	8
例 6	宛先のプリフィックスが9ならば、プリフィックス8を付けたユーザー名でhostにルーティングする。...	8
例 7	宛先のユーザーがレジストされていなければ、sip:user@hostにルーティングする。.....	8
例 8	発信者のユーザー名が adminならば、sip:user@hostにルーティングする。.....	8
例 9	発信者がレジストされていなければ、sip:user@hostにルーティングする。.....	8
例 10	発信元IPアドレスが 192.168.0.1 ならば、sip:user@hostにルーティングする。.....	8
2.2.	ルーティング先としてユーザー名を指定.....	9
例 11	ユーザーuserがレジストされていれば、ユーザーuserにルーティングする。.....	9
例 12	宛先のユーザーがレジストされていなければ、ユーザーuserにルーティングする。.....	9
例 13	宛先のユーザー名にプリフィックス9を付けてルーティングする。.....	9
例 14	時刻が午前10時から午後5時59分の時間内ならば、ユーザーuserにルーティングする。.....	9
例 15	日付が12月12日から19日の期間内ならば、ユーザーuserにルーティングする。.....	9
2.3.	ルーティング先としてIPアドレス/FQDNを指定.....	10
例 16	宛先のホスト名が hostならば、アドレスserverにルーティングする。.....	10
例 17	発信元IPアドレスが 192.168.0.1 ならば、アドレスhost.domainにルーティングする。.....	10
例 18	発信元がローカルホストならば、アドレスhost.domainにルーティングする。.....	10
例 19	発信元ポート番号が 15060 ならば、アドレスhost.domainにルーティングする。.....	10
例 20	リクエストのメソッドがSUBSCRIBEならば、192.168.0100 にルーティングする。.....	10
3.	リジェクト	11
例 21	宛先のユーザーがレジストされていなければ、レスポンス 603 Declineを返す。.....	11
例 22	宛先のSIP URI がsip:user@hostならば、レスポンス 486 Busyを返す。.....	11
例 23	宛先のユーザー名のプリフィックスが9ならば、レスポンス 402 Payment Requiredを返す。.....	11
例 24	発信者のユーザー名が userならば、レスポンス 404 Not Foundを返す。.....	11
例 25	発信元IPアドレスのプリフィックスが 192.168 ならば、レスポンス 403 Forbiddenを返す。.....	11
例 26	Content-Typeが application/text ならば、レスポンス 406 Not Acceptableを返す。.....	12
例 27	User-Agentの値が文字列 TESTを含むならば、レスポンス 503 Service Unavailableを返す。.....	12
例 28	Max-Forwardsの値が5以下ならば、レスポンス 483 Too Many Hopsを返す。.....	12
例 29	時刻が午前0時から午前7時59分の時間内ならば、レスポンス 480 Temporarily Unavailableを返す。.....	12

す。	12
例 30	リクエストのメソッドがSUBSCRIBEならば、レスポンス 400 Bad Requestを返す。..... 12
4.	SIPヘッダの編集13
4.1.	既存のSIPヘッダを書き換え..... 13
例 31	発信者のユーザー名が adminならば、ディスプレイネームを "Ted"にする。..... 13
例 32	Expiresの値が 199 以下ならば、その値を 200 にする。..... 13
例 33	User-Agentの値が文字列 Alphaを含むならば、その値を Betaにする。..... 13
4.2.	SIPヘッダの追加 14
例 34	X-Exampleヘッダを追加する。..... 14
4.3.	SIPヘッダの削除 14
例 35	User-Agentヘッダを削除する。..... 14
5.	認証15
例 36	宛先のFQDNが host.domain ならば、認証を要求する。..... 15
例 37	宛先のユーザーのプリフィックスが 800 ならば、認証を要求しない。..... 15
例 38	発信者がレジストされていないならば、認証を要求する。..... 15
例 39	発信元IPアドレスのプリフィックスが 192.168.10 ならば、認証を要求する。..... 15
例 40	時刻が午前 10 時から午前 5 時 59 分の時間内ならば、認証を要求しない。..... 16
6.	負荷の分散17
例 41	1秒おきにルーティング先を3カ所に切り替える。..... 17
例 42	30 分おきにルーティング先を2カ所に切り替える。..... 17
例 43	セッションIDの偶数・奇数でルーティング先を切り替える。..... 17
7.	NATトラバーサル.....18
7.1.	NATトラバーサルの有効／無効..... 18
例 44	宛先のFQDNが host.domain ならば、NATトラバーサル機能を有効にする。..... 18
例 45	発信元IPアドレスが 192.168.0.1 ならば、NATトラバーサル機能を無効にする。..... 18
7.2.	インターフェース・アドレスの指定..... 19
例 46	宛先のコンタクト先IPアドレスのプリフィックスが 192.168.1 ならば、インターフェース・アドレスとして 192.168.1.1 を使用する。..... 19
例 47	発信元IPアドレスのプリフィックスが 192.168.2 ならば、インターフェース・アドレスとして 192.168.2.1 を使用する。..... 19
8.	RTPリレー20
例 48	宛先のユーザー名のプリフィックスが 9 ならば、RTPリレー機能を有効にする。..... 20
例 49	宛先のSIP URIが sip:user@hostならば、RTPリレー機能を有効にする。このときペイロード・タイプを

PCMAにする。.....	20
例 50 発信元IPアドレスが 192.168.0.1 ならば、RTPリレー機能を有効にする。このときRTPポートの範囲を 10000 から 10100 にする。.....	20
9. 環境変数の設定	21
例 51 全ての通話でコーデックとしてG723を使う。.....	21
例 52 宛先のユーザー名のプリフィックスが 9 ならば、Record-Routeヘッダを付けない。.....	21
例 53 宛先のホスト名が hostならば、Record-Routeヘッダに lrパラメータを付けない。.....	21
例 54 宛先のホスト名が hostならば、Viaヘッダにrportパラメータを付けない。.....	21
例 55 発信者のユーザー名が adminならば、リングング・タイムアウトを 30 秒にする。.....	21
例 56 発信者のユーザー名のプリフィックスが 9 ならば、host.domainにUpper Registrationする。.....	21
例 57 レジスタのExpires値が 99 以下であれば、以降のレジスタ間隔を 100 秒にする。.....	22
例 58 宛先のホスト名が hostならば、Thru Registrationしない。.....	22
10. セッション・プラグインのロード.....	23
例 59 全ての通話でプラグインRadiusAcctを使う。.....	23
例 60 宛先のホスト名が hostならば、CDRlogプラグインを使う。.....	23

1. はじめに

本ドキュメントでは、OnDO SIP Server のダイヤル・プランについてサンプルを挙げています。

ユーザーの目的に合わせて変更の上、利用されることを想定しています。

ダイヤル・プランの基本、文法 及び 管理方法については「OnDO SIP Server アドミニストレータ・ガイド」の「6. ダイヤル・プラン」を参照して下さい。

本ドキュメントで紹介している機能は以下の通りです。

- ◆ ルーティング
- ◆ リジェクト
- ◆ SIP ヘッダの編集
- ◆ 認証
- ◆ 負荷の分散
- ◆ NATトラバーサル
- ◆ RTP リレー
- ◆ 環境変数の設定
- ◆ セッション・プラグインのロード

2. ルーティング

ルーティングはダイヤル・プランの主な目的です。設定方法として以下の3つがあり、いずれも Deploy Patterns で定義されます。これらの設定内容是对应する Matching Patterns の条件が満たされたときのみ有効になります。

To = ルーティング先の SIP URI

例: To = sip:user@host

アドレス host にセッションをルーティングします。

To = ルーティング先のユーザー名

例: To = sip:user@

ユーザー 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。

コンタクト先アドレスはレジスタ時に登録されたアドレスです。

\$target = ルーティング先の SIP URI または FQDN または IP アドレス

例: `$target = sip:user@host`

`$target = host`

アドレス host にセッションをルーティングします。

`$target = 192.168.0.10`

アドレス 192.168.0.10 にセッションをルーティングします。

2.1. ルーティング先として SIP URI を指定する場合

例1 全ての通話を `sip:user@host` にルーティングする。

Matching Patterns	Deploy Patterns
<code>\$request = ^INVITE</code>	<code>To = sip:user@host</code>

例2 宛先のユーザー名が admin ならば、`sip:user@host` にルーティングする。

Matching Patterns	Deploy Patterns
<code>\$request = ^INVITE</code> <code>To = sip:admin@</code>	<code>To = sip:user@host</code>

例3 宛先の SIP URI が `sip:admin@server` ならば、`sip:user@host` にルーティングする。

Matching Patterns	Deploy Patterns
<code>\$request = ^INVITE</code> <code>\$geturi(To) = sip:admin@server</code>	<code>To = sip:user@host</code>

例4 宛先のユーザー名のプリフィックスが 9 ならば、同じユーザー名のまま host にルーティングする。

Matching Patterns	Deploy Patterns
<code>\$request = ^INVITE</code> <code>To = sip:(9.+)</code>	<code>To = sip:%1@host</code>

- 例5 宛先のプリフィックスが9ならば、プリフィックスを削除したユーザー名で host にルーティングする。

Matching Patterns	Deploy Patterns
<pre>\$request = ^INVITE To = sip:9(.+)</pre>	<pre>To = sip:%1@host</pre>

- 例6 宛先のプリフィックスが9ならば、プリフィックス8を付けたユーザー名で host にルーティングする。

Matching Patterns	Deploy Patterns
<pre>\$request = ^INVITE To = sip:(9.+)</pre>	<pre>To = sip:8%1@host</pre>

- 例7 宛先のユーザーがレジストされていなければ、sip:user@host にルーティングする。

Matching Patterns	Deploy Patterns
<pre>\$request = ^INVITE \$registered = false</pre>	<pre>To = sip:user@host</pre>

- 例8 発信者のユーザー名が admin ならば、sip:user@host にルーティングする。

Matching Patterns	Deploy Patterns
<pre>\$request = ^INVITE From = sip:admin@</pre>	<pre>To = sip:user@host</pre>

- 例9 発信者がレジストされていなければ、sip:user@host にルーティングする。

Matching Patterns	Deploy Patterns
<pre>\$request = ^INVITE \$registered(From) = false</pre>	<pre>To = sip:user@host</pre>

- 例10 発信元 IP アドレスが 192.168.0.1 ならば、sip:user@host にルーティングする。

Matching Patterns	Deploy Patterns
<pre>\$request = ^INVITE \$addr = 192¥.168¥.0¥.1¥</pre>	<pre>To = sip:user@host</pre>

2.2. ルーティング先としてユーザー名を指定

例11 ユーザーuser がレジストされていれば、ユーザーuser にルーティングする。

Matching Patterns	Deploy Patterns
<pre>\$request = ^INVITE \$registered("user") = true</pre>	<pre>To = sip:user@</pre>

例12 宛先のユーザーがレジストされていなければ、ユーザーuser にルーティングする。

Matching Patterns	Deploy Patterns
<pre>\$request = ^INVITE \$registered = false \$registered("user") = true</pre>	<pre>To = sip:user@</pre>

例13 宛先のユーザー名にプリフィックス 9 を付けてルーティングする。

Matching Patterns	Deploy Patterns
<pre>\$request = ^INVITE To = sip:(.+)@</pre>	<pre>To = sip:9%1@</pre>

例14 時刻が午前 10 時から午後 5 時 59 分の時間内ならば、ユーザーuser にルーティングする。

Matching Patterns	Deploy Patterns
<pre>\$request = ^INVITE \$time = ^1[0-7]:</pre>	<pre>To = sip:user@</pre>

例15 日付が 12 月 12 日から 19 日の期間内ならば、ユーザーuser にルーティングする。

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

2.3. ルーティング先として IP アドレス/FQDN を指定

例16 宛先のホスト名が host ならば、アドレス server にルーティングする。

Matching Patterns	Deploy Patterns
<pre>\$request = ^INVITE \$geturi(To) = @host</pre>	<pre>\$target = server</pre>

例17 発信元 IP アドレスが 192.168.0.1 ならば、アドレス host.domain にルーティングする。

Matching Patterns	Deploy Patterns
<pre>\$request = ^INVITE \$addr = 192¥.168¥.0¥.1¥</pre>	<pre>\$target = host.domain</pre>

例18 発信元がローカルホストならば、アドレス host.domain にルーティングする。

Matching Patterns	Deploy Patterns
<pre>\$request = ^INVITE \$localhost = true</pre>	<pre>\$target = host.domain</pre>

例19 発信元ポート番号が 15060 ならば、アドレス host.domain にルーティングする。

Matching Patterns	Deploy Patterns
<pre>\$request = ^INVITE \$port = ^15060¥</pre>	<pre>\$target = 192.168.0.100</pre>

例20 リクエストのメソッドが SUBSCRIBE ならば、192.168.0.100 にルーティングする。

Matching Patterns	Deploy Patterns
<pre>\$request = ^SUBSCRIBE</pre>	<pre>\$target = 192.168.0.100</pre>

3. リジェクト

ここではリジェクトの方法として、エラー・レスポンスを返す例を挙げています。

エラー・レスポンスは処理変数\$action に SIP のレスポンス・コード番号を指定することによってリクエストの送信者に送られます。エラー・レスポンスが返されたセッションはいかなる宛先にもルーティングされません。

レスポンス・コード番号については RFC3261 等を参照して下さい。

例21 宛先のユーザーがレジストされていなければ、レスポンス 603 Decline を返す。

Matching Patterns	Deploy Patterns
<pre>\$request = ^INVITE \$registered = false</pre>	<pre>\$action = 603</pre>

例22 宛先の SIP URI が sip:user@host ならば、レスポンス 486 Busy を返す。

Matching Patterns	Deploy Patterns
<pre>\$request = ^INVITE \$geturi(To) = sip:user@host</pre>	<pre>\$action = 486</pre>

例23 宛先のユーザー名のプリフィックスが 9 ならば、レスポンス 402 Payment Required を返す。

Matching Patterns	Deploy Patterns
<pre>\$request = ^INVITE To = sip:9.+@</pre>	<pre>\$action = 402</pre>

例24 発信者のユーザー名が user ならば、レスポンス 404 Not Found を返す。

Matching Patterns	Deploy Patterns
<pre>\$request = ^INVITE From = sip:user@</pre>	<pre>\$action = 404</pre>

例25 発信元 IP アドレスのプリフィックスが 192.168 ならば、レスポンス 403 Forbidden を返す。

Matching Patterns	Deploy Patterns
<pre>\$request = ^INVITE \$addr = 192¥.168¥.</pre>	<pre>\$action = 403</pre>

例26 Content-Type が application/text ならば、レスポンス 406 Not Acceptable を返す。

Matching Patterns	Deploy Patterns
<pre>\$request = ^INVITE Content-Type=^application/text\$</pre>	<pre>\$action = 406</pre>

例27 User-Agent の値が文字列 TEST を含むならば、レスポンス 503 Service Unavailable を返す。

Matching Patterns	Deploy Patterns
<pre>\$request = ^INVITE User-Agent = TEST</pre>	<pre>\$action = 503</pre>

例28 Max-Forwards の値が5以下ならば、レスポンス 483 Too Many Hops を返す。

Matching Patterns	Deploy Patterns
<pre>\$request = ^INVITE Max-Forwards = ^[0-5]\$</pre>	<pre>\$action = 483</pre>

例29 時刻が午前 0 時から午前 7 時 59 分の時間内ならば、レスポンス 480 Temporarily Unavailable を返す。

Matching Patterns	Deploy Patterns
<pre>\$request = ^INVITE \$time = ^0[0-7]:</pre>	<pre>\$action = 480</pre>

例30 リクエストのメソッドが SUBSCRIBE ならば、レスポンス 400 Bad Request を返す。

Matching Patterns	Deploy Patterns
<pre>\$request = ^SUBSCRIBE</pre>	<pre>\$action = 400</pre>

4. SIP ヘッダの編集

SIP ヘッダの編集とはダイヤル・プランによるヘッダの置換、追加、削除を意味します。

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、存在しなければヘッダが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。

特定の SIP ヘッダを編集することによって、発側に関する情報の漏洩を防いだり、着側に対する相互接続性を高めたりすることができます。

4.1. 既存の SIP ヘッダを書き換え

例31 発信者のユーザー名が admin ならば、ディスプレイネームを “Ted” にする。

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

例32 Expires の値が 199 以下ならば、その値を 200 にする。

Matching Patterns	Deploy Patterns
<pre>\$request = ^INVITE Expires = ^[01][0-9].\$</pre>	<pre>Expires = 200</pre>

例33 User-Agent の値が文字列 Alpha を含むならば、その値を Beta にする。

Matching Patterns	Deploy Patterns
<pre>\$request = ^INVITE User-Agent = (.*)Alpha(.*)</pre>	<pre>User-Agent = %1Beta%2</pre>

4.2. SIP ヘッダの追加

例34 X-Example ヘッダを追加する。

Matching Patterns	Deploy Patterns
<code>\$request = ^INVITE</code>	<code>X-Example = hello</code>

4.3. SIP ヘッダの削除

例35 User-Agent ヘッダを削除する。

Matching Patterns	Deploy Patterns
<code>\$request = ^INVITE</code>	<code>User-Agent =</code>

5. 認証

ダイヤル・プランでは特定のリクエストに対して認証を求めることができます。逆にシステムの設定で認証要求が有効であったとしても、特定のリクエストに対しては認証を無効にすることもできます。これらは処理変数 `$auth` によって設定され、値が `true` の場合は有効、`false` の場合は無効です。

認証が有効である場合、OnDO SIP Server はリクエストに対してレスポンス 407 Proxy Authentication Required (または 401 Unauthorized)を返し、認証情報を伴ったリクエストの再送を促します。

OnDO SIP Server は、リクエストが正しい認証情報を含んでいない限り、そのリクエストを受け入れることはありません。

例36 宛先の FQDN が `host.domain` ならば、認証を要求する。

Matching Patterns	Deploy Patterns
<pre>\$request = ^INVITE \$geturi(To) = @host.domain</pre>	<pre>\$auth = true</pre>

例37 宛先のユーザーのプリフィックスが `800` ならば、認証を要求しない。

Matching Patterns	Deploy Patterns
<pre>\$request = ^INVITE To = sip:800.+@</pre>	<pre>\$auth = false</pre>

例38 発信者がレジストされていなければ、認証を要求する。

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

例39 発信元 IP アドレスのプリフィックスが `192.168.10` ならば、認証を要求する。

Matching Patterns	Deploy Patterns
<pre>\$request = ^INVITE \$addr = 192¥.168¥.¥10</pre>	<pre>\$auth = true</pre>

例40 時刻が午前 10 時から午前 5 時 59 分の時間内ならば、認証を要求しない。

Matching Patterns	Deploy Patterns
<pre>\$request = ^INVITE \$time = ^1[0-7]:</pre>	<pre>\$auth = false</pre>

6. 負荷の分散

ダイヤル・プランでは、時間やセッション ID などの特定の値を基にルーティング先を分散させることができます。これにより、たとえば VoIP ゲートウェイなどを複数用意することによって、1 つのゲートウェイに負荷が集中することを回避できます。

例41 1秒おきにルーティング先を3カ所に切り替える。

Matching Patterns	Deploy Patterns
<code>\$request = ^INVITE</code> <code>\$time = [0369]\$</code>	<code>\$target = server1</code>
<code>\$request = ^INVITE</code> <code>\$time = [147]\$</code>	<code>\$target = server2</code>
<code>\$request = ^INVITE</code>	<code>\$target = server3</code>

例42 30分おきにルーティング先を2カ所に切り替える。

Matching Patterns	Deploy Patterns
<code>\$request = ^INVITE</code> <code>\$time = ^..:[0-2]</code>	<code>\$target = server1</code>
<code>\$request = ^INVITE</code>	<code>\$target = server2</code>

例43 セッション ID の偶数・奇数でルーティング先を切り替える。

Matching Patterns	Deploy Patterns
<code>\$request = ^INVITE</code> <code>\$sid = [13579]\$</code>	<code>\$target = server1</code>
<code>\$request = ^INVITE</code>	<code>\$target = server2</code>

7. NATトラバースル

OnDO SIP Server はセッションごとに NATトラバースルの必要性を自動的に判断しますが、管理者がダイヤル・プランによって指定することもできます。

この章では以下の処理変数を使用します。

\$nat = true または false

NATトラバースル処理の有無

\$ifdst = IP アドレス または FQDN

送信先(着側)方向のインターフェース・アドレス

このアドレスは、Via:、Record-Route: などに使用されます。

\$ifsrc = IP アドレス または FQDN

送信元(発側)方向のインターフェース・アドレス

このアドレスは、Via:、Record-Route: などに使用されます。

7.1. NATトラバースルの有効／無効

例44 宛先の FQDN が host.domain ならば、NATトラバースル機能を有効にする。

Matching Patterns	Deploy Patterns
<pre>\$request = ^INVITE \$geturi(To) = @host.domain</pre>	<pre>\$nat = true</pre>

例45 発信元 IP アドレスが 192.168.0.1 ならば、NATトラバースル機能を無効にする。

Matching Patterns	Deploy Patterns
<pre>\$request = ^INVITE \$addr = 192¥.168¥.0¥.1¥</pre>	<pre>\$nat = false</pre>

7.2. インターフェース・アドレスの指定

例46 宛先のコンタクト先 IP アドレスのプリフィックスが 192.168.1 ならば、インターフェース・アドレスとして 192.168.1.1 を使用する。

Matching Patterns	Deploy Patterns
<pre>\$request = ^INVITE \$registeredaddr = ^192¥.168¥.1¥.</pre>	<pre>\$ifdst = 192.168.0.1.1</pre>

例47 発信元 IP アドレスのプリフィックスが 192.168.2 ならば、インターフェース・アドレスとして 192.168.2.1 を使用する。

Matching Patterns	Deploy Patterns
<pre>\$request = ^INVITE \$addr = ^192¥.168¥.2¥.</pre>	<pre>\$ifsrc = 192.168.0.2.1</pre>

8. RTP リレー

OnDO SIP Server はセッションごとに RTP リレーの必要性を自動的に判断しますが、管理者がダイヤル・プランによって指定することもできます。NAT トラバーサルが行われているセッションでは、自動的に RTP リレーも行われます。

この章では以下の処理変数を使用します。

`$rtp = true` または `false`

RTP リレーの有無

`$ifdst = IP アドレス` または `FQDN`

送信先(着側)方向のインターフェース・アドレス

着側から送られる RTP パケットの待ち受けアドレスとして使われます。

`$ifsrc = IP アドレス` または `FQDN`

送信元(発側)方向のインターフェース・アドレス

発側から送られる RTP パケットの待ち受けアドレスとして使われます。

例48 宛先のユーザー名のプリフィックスが 9 ならば、RTP リレー機能を有効にする。

Matching Patterns	Deploy Patterns
<pre>\$request = ^INVITE To = sip:9.+@</pre>	<pre>\$rtp = true</pre>

例49 宛先の SIP URI が `sip:user@host` ならば、RTP リレー機能を有効にする。このときペイロード・タイプを PCMA にする。

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

例50 発信元 IP アドレスが `192.168.0.1` ならば、RTP リレー機能を有効にする。このとき RTP ポートの範囲を 10000 から 10100 にする。

Matching Patterns	Deploy Patterns
<pre>\$request = ^INVITE \$addr = 192¥.168¥.0¥.1¥</pre>	<pre>\$rtp = true &net.rtp.port.min = 10000 &net.rtp.port.max = 10100</pre>

9. 環境変数の設定

環境変数はサーバの動作や管理情報、各種内部パラメータを設定するための変数です。

ダイヤル・プランによって環境変数を指定する場合は、変数名のプリフィックスとして & を付けてください。また、設定された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。

例51 全ての通話でコーデックとして G723 を使う。

Matching Patterns	Deploy Patterns
<code>\$request = ^INVITE</code>	<code>&net.rtp.audio.payloadtype = 4</code>

例52 宛先のユーザー名のプリフィックスが 9 ならば、Record-Route ヘッダを付けない。

Matching Patterns	Deploy Patterns
<code>\$request = ^INVITE</code> <code>To = sip:9.+@</code>	<code>&net.sip.addreordroute = false</code>

例53 宛先のホスト名が host ならば、Record-Route ヘッダに lr パラメータを付けない。

Matching Patterns	Deploy Patterns
<code>\$request = ^INVITE</code> <code>\$geturi(To) = @host</code>	<code>&net.sip.addreordroute.lr = false</code>

例54 宛先のホスト名が host ならば、Via ヘッダに rport パラメータを付けない。

Matching Patterns	Deploy Patterns
<code>\$request = ^INVITE</code> <code>\$geturi(To) = @host</code>	<code>\$net.sip.addrport = false</code>

例55 発信者のユーザー名が admin ならば、リングング・タイムアウトを 30 秒にする。

Matching Patterns	Deploy Patterns
<code>\$request = ^INVITE</code> <code>From = sip:admin@</code>	<code>\$net.sip.timeout.ringing = 30000</code>

例56 発信者のユーザー名のプリフィックスが 9 ならば、host.domain に Upper Registration する。

Matching Patterns	Deploy Patterns
-------------------	-----------------

<pre>\$request = ^REGISTER From = sip:9.+@</pre>	<pre>&net.registrar.upper.allow = true &net.registrar.upper.url = host.domain \$continue = true</pre>
--	---

例57 レジスタの Expires 値が 99 以下であれば、以降のレジスタ間隔を 100 秒にする。

Matching Patterns	Deploy Patterns
<pre>\$request = ^REGISTER Expires = ^[0-9].\$</pre>	<pre>&net.registrar.adjust.expires=100 \$continue = true</pre>

例58 宛先のホスト名が host ならば、Thru Registration しない。

Matching Patterns	Deploy Patterns
<pre>\$request = ^REGISTER \$geturi(To) = @host</pre>	<pre>&net.registrar.thru.allow = false \$continue = true</pre>

10. セッション・プラグインのロード

セッション・プラグインはセッション制御や課金情報収集などに使われるプラグインです。処理変数 `$session` にプラグイン名を設定することによってロードされます。

例59 全ての通話でプラグイン RadiusAcct を使う。

Matching Patterns	Deploy Patterns
<code>\$request = ^INVITE</code>	<code>\$session=com.sample.radius.proxy.RadiusAcct</code>

例60 宛先のホスト名が host ならば、CDRlog プラグインを使う。

Matching Patterns	Deploy Patterns
<code>\$request = ^INVITE</code> <code>\$geturi(To) = @host</code>	<code>\$session = com.user.CDRlog</code>