# inGate

# How To Guide

## Regular Expressions in the Dial Plan: UK edition
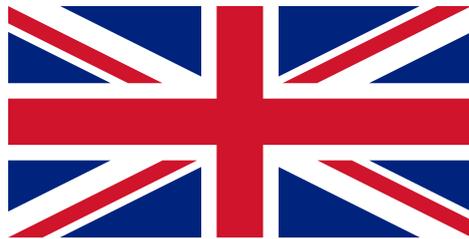
08 May 2009

# Table of Contents

Tested versions:          Ingate Firewall and SIParator version 4.7.1

Revision History:

| Revision | Date | Author | Comments |
| --- | --- | --- | --- |
| 1 | 2009-05-08 | Paul Donald | First Public Release |

Based on Regular Expressions in the dial-plan document by Scott Beer

# Regular Expressions and the SIP Protocol

Within the SIP Protocol (RFC 3261), a SIP URI identifies a communications resource. Like all URIs, SIP URIs may be placed in web pages, email messages, or printed literature. They contain sufficient information to initiate and maintain a communication session with the resource. In its simplest form a SIP URI looks like `sip:user@host`, where the "`user`" is the identifier of a particular client resource at the host being addressed. The term "host" in this context frequently refers to a domain, a network location.

> When using Regular Expressions (RegExps) to define a SIP URI, these expressions allow for the customization and flexibility to define SIP URIs as variables. A powerful tool when used, the regular expression can increase the capabilities of the Ingate Dial Plan as it further increases the definitions of the SIP URIs used to route calls. The Regular Expression can be used to match and define both the user and host portion of the SIP URI.
>
> When designing a RegExp, spare some thought for UK standard telecomm dialling notation; that local numbers not begin with a zero, national numbers do, and international numbers be prefixed with 00.

Industry-standard regular expression notations follow – note that character matches are `CaSe SenSitivE`:

| Operator | Description |
|---|---|
| `[abc]` | Matches any one character in the set `a`, `b`, or `c` |
| `[^abc]` | Matches any one character not in the set `a`, `b`, or `c` |
| `[a-z]` | Matches any one character in the range `a-z` but not `A-Z` |
| `[1-8]` | Matches any one character "one through eight inclusive" |
| `[369]` | Matches any one character "3, or 6, or 9" |
| `.` | Matches any one single alphanumeric character |
| `,` | Matches the minimum specified characters or more |
| `\d` | Matches any decimal digit |
| `\D` | Matches any non-digit |
| `\s` | Matches any whitespace character |
| `\S` | Matches any non-whitespace character |
| `\w` | Matches any word (alphanumeric) character |
| `\W` | Matches any non-word (alphanumeric) character |
| `(abc)` | Matches the sequence `abc` , stores it as a variable which may be used in later expressions - also used for grouping |
| `$1` | The $ symbol is used to recall expressions that have been stored within `(` and `)` above - variables are numbered according to the capture heirarchy |
| `$2` | The second stored variable |
| `a\|b` | Matches `a` , or `b` |
| `+` | Matches the preceding expression one or more times |
| `?` | Makes *preceding* item *optional*; if the preceding is inside `()` brackets, e.g. `(345)?` then that `345` is optional |
| `*` | Matches a null string or any number of repetitions of the preceding expression |
| `{m}` | Matches exactly `m` repetitions of the one-character expression |
| `{m,n}` | Matches `m` to `n` repetitions of the preceding expression |
| `{m,}` | Matches `m` or more repetitions of the preceding expression. |

## Routing Calls using the Dial Plan

The dial plan is used to instruct the Ingate SIP Proxy or B2BUA about how to route calls, who to accept calls from, match on where calls are going to and where to send calls. Adding regular expressions into the dial plan allows you to specify a range of numbers, range of domains, or other sets of specific alphanumerics.

> Regular expressions are a flexible way of delivering patterns that match a unique set of criteria. For example, if you specify the regular expression `[0-9]{7,}`, the Ingate Firewall/SIParator recognizes any digits from zero to nine, repeated seven or more times, in other words, a telephone number.

## Regular Expressions in the Matching From Header

The purpose of the Matching From Header is to provide source SIP URI matching, i.e. matching the FROM: field or the origin of a call.

| SIP URI Example Description | Equivalent Regular Expression |
|---|---|
| 7-digit number@ Any Domain<br>7-digit number@ IP Address<br>7-digit number@ Domain<br>7-digit number@ (anything hereafter doesn't matter) | `sip:[0-9]{7}@(.*)`<br>`sip:[0-9]{7}@12.34.56.78`<br>`sip:[0-9]{7}@sip_domain.com`<br>`sip:[0-9]{7}@` |
| 7-digit numbers within the London area codes 0207 and 0208 @ | `sip:020[78][0-9]{7}@` |
| 7-digit number with 0845 Lo-call prefix @<br>6 to 7-digit number with 0845 Lo-call prefix @<br>6-digit number with 0845 Lo-call prefix @ | `sip:0845[0-9]{7}@`<br>`sip:0845[0-9]{6,7}@`<br>`sip:0845[0-9]{6}@` |
| 7-digit number with 0870 National prefix @<br>6 to 7-digit number with 0870 National prefix @<br>6-digit number with 0870 National prefix @ | `sip:0870[0-9]{7}@`<br>`sip:0870[0-9]{6,7}@`<br>`sip:0870[0-9]{6}@` |
| 7-digit number with 0845 Lo-call or 0870 National prefix @<br>Note: the first can also match 0875 or 0840, the second wont | `sip:08[74][05][0-9]{7}@`<br>`sip:(0870\|0845)[0-9]{7}@` |
| Local 7-digit number, beginning with optional 9 @ any alphanumeric domain string | `sip:9?[0-9]{7}@(.*)` |
| 4-digit number (i.e. an extension): first digit is 5, then any 3 digits @ any alphanumeric domain string | `sip:5[0-9]{3}@(.*)` |
| 4-digit number *not* starting with 36 @ | `sip:(?!36)[0-9]{4}@` |
| Any length alphanumeric username @ any alphanumeric domain string | `sip:(.*)@(.*)` |
| Any 9 to 10-digit UK numbers prefixed with optional 00 and then mandatory 44 | `sip:(00)?44[0-9]{9,10}@` |
| 7-digit Liverpool 151 numbers prefixed with optional 00 and then mandatory 44 | `sip:(00)?44151[0-9]{7}@` |

| | |
|---|---|
| 7-digit Liverpool 151 numbers prefixed with 0 | `sip:0151[0-9]{7}@` |
| 7-digit Birmingham 151 numbers prefixed with 0 | `sip:0121[0-9]{7}@` |
| 6 to 7-digit Coventry 2476 numbers prefixed with 0 | `sip:02476[0-9]{6,7}@` |
| 6 to 7-digit Manchester 161 numbers prefixed with 0 | `sip:0161[0-9]{6,7}@` |
| 7-digit Leicester 116 numbers prefixed with 0 | `sip:0116[0-9]{7}@` |
| 6-digit mobile 7989 numbers prefixed with 0 | `sip:07989[0-9]{6}@` |
| Any 9-digit mobile 7 number prefixed with 0 | `sip:07[0-9]{9}@` |
| optional 353 prefix with or without optional 00 start, then optional 0 with optional 1-2 digit area code, then mandatory 7-digit number @ any domain.<br>Note that the following URI's will match:<br>sip:0035312345678@abc.com<br>sip:35312345678@asdf.com<br>sip:12345678@asdf .com<br>sip:2345678@asdf .com<br>sip:012345678@domain.com<br>sip:0212345678@wherever | `sip:(((00)?353)?0?[0-9]{1,2})?([0-9]{7})@(.*)` |
| optional 44 prefix with or without optional 00 start, then optional 0 in cases of no national code, then optional 1-3 digit area code, then mandatory 7-digit number @ any domain.<br>Note that the following URI's will match:<br>sip:004412345678@abc.com<br>sip:4412345678@asdf.com<br>sip:12345678@asdf .com<br>sip:2345678@asdf .com<br>sip:012345678@domain.com<br>sip:01512345454@wherever | `sip:(((00)?44)?0?[0-9]{1,3})?([0-9]{7})@(.*)` |

## *Regular Expressions in the Matching Request-URI*

The purpose of the Matching Request-URI is to match a Request URI Header of the SIP messages being sent through the Ingate and determine where it wants to go.

Typically the "domain" portion of the URI is the Ingate IP Address or FQDN. In the Regular Expression, you create criteria for the Request-URI of the SIP messages. This is used to match requests in the Dial Plan. For a request to match, all criteria must be fulfilled. Port and Transport can be used but will only match if there are both port and transport parameters contained in the SIP URI.

| SIP URI Example Description | Equivalent Regular Expression |
|---|---|
| 7-digit number@ Any Domain<br>7-digit number@ IP Address<br>7-digit number@ Domain<br>7-digit number@ (anything hereafter doesn't matter) | `sip:[0-9]{7}@(.*)`<br>`sip:[0-9]{7}@12.34.56.78`<br>`sip:[0-9]{7}@sip_domain.com`<br>`sip:[0-9]{7}@` |
| Emergency numbers 112 or 999 @<br><br>*Warning: not all SIP providers have access to emergency service numbers.* | `sip:112|999@` |
| 7-digit numbers within the London area codes 0207 and 0208 @ | `sip:020[78][0-9]{7}@` |
| 7-digit number with 0845 Lo-call prefix @<br>6 to 7-digit number with 0845 Lo-call prefix @<br>6-digit number with 0845 Lo-call prefix @ | `sip:0845[0-9]{7}@`<br>`sip:0845[0-9]{6,7}@`<br>`sip:0845[0-9]{6}@` |
| 7-digit number with 0870 National prefix @<br>6 to 7-digit number with 0870 National prefix @<br>6-digit number with 0870 National prefix @ | `sip:0870[0-9]{7}@`<br>`sip:0870[0-9]{6,7}@`<br>`sip:0870[0-9]{6}@` |
| 7-digit number with 0845 Lo-call or 0870 National prefix @<br><br>Note: the first can also match 0875 or 0840, the second wont | `sip:08[74][05][0-9]{7}@`<br>`sip:(0870|0845)[0-9]{7}@` |
| Local 7-digit number, beginning with optional 9 @ any alphanumeric domain string | `sip:9?[0-9]{7}@(.*)` |
| Remove Prefix "1613" on any username @ Anything<br>Remove Prefix "1613" on any username @ IP Address<br>Remove Prefix "1613" on any username @ Domain<br><br>$1 is `(.*)` i.e. 1613 is matched but not stored. If the number doesn't begin "1613", there will be no match. | `sip:1613(.*)@(.*)`<br>`sip:1613(.*)@12.34.56.78`<br>`sip:1613(.*)@sip_domain.com` |
| Remove Prefix "1613" on 11-digit number@ Anything<br>Remove Prefix "1613" on 11-digit number@ IP Address<br>Remove Prefix "1613" on 11-digit number@ Domain<br><br>and pass only 7-digits<br><br>$1 is `([0-9]{7})`   $2 is `(.*)` | `sip:1613([0-9]{7})@(.*)`<br>`sip:1613([0-9]{7})@12.34.56.78`<br>`sip:1613([0-9]{7})@sip_domain.com` |
| Remove optional "+" Prefix on Any Username @ Anything<br>Remove optional "+" Prefix on Any Username @ IP Address<br>Remove optional "+" Prefix on Any Username @ Domain<br><br>note: as you've already read, \+ (the literal + character) has a different meaning to + (a regexp processing rule). | `sip:\+?(.*)@(.*)`<br>`sip:\+?(.*)@12.34.56.78`<br>`sip:\+?(.*)@sip_domain.com` |

| | |
|---|---|
| Any Username @ Any Domain<br>Any Username @ IP Address<br>Any Username @ Domain | `sip:(.*)@(.*)`<br>`sip:(.*)@12.34.56.78`<br>`sip:(.*)@sip_domain.com` |
| Any Username @ Any Domain with Port 5060<br>Any Username @ IP Address with Port 5060<br>Any Username @ Domain with Port 5060 | `sip:(.*)@(.*):5060`<br>`sip:(.*)@12.34.56.78:5060`<br>`sip:(.*)@sip_domain.com:5060` |
| Any Username @ Any Domain with Port and Transport<br>Any Username @ IP Address with Port and Transport<br>Any Username @ Domain with Port and Transport | `sip:(.*)@(.*):5060;transport=UDP`<br>`sip:(.*)@12.34.56.78:5060;transport=UDP`<br>`sip:(.*)@sip_domain.com:5060;transport=UDP` |
| 6 to 7-digit Coventry 2476 numbers prefixed with 0 | `sip:02476[0-9]{6,7}@` |
| 6 to 7-digit Manchester 161 numbers prefixed with 0 | `sip:0161[0-9]{6,7}@` |
| 7-digit Leicester 116 numbers prefixed with 0 | `sip:0116[0-9]{7}@` |
| 6-digit mobile 7989 numbers prefixed with 0 | `sip:07989[0-9]{6}@` |
| Any 9-digit mobile 7 number prefixed with 0 | `sip:07[0-9]{9}@` |
| optional 353 prefix with or without optional 00 start, then optional 0 with optional 1-2 digit area code, then mandatory 7-digit number @ any domain.<br>Note that the following URI's will match:<br>sip:0035312345678@abc.com<br>sip:35312345678@asdf.com<br>sip:12345678@asdf .com<br>sip:2345678@asdf .com<br>sip:012345678@domain.com<br>sip:0212345678@wherever | `sip:(((00)?353)?0?[0-9]{1,2})?([0-9]{7})@(.*)` |
| optional 44 prefix with or without optional 00 start, then optional 0 in cases of no national code, then optional 1-3 digit area code, then mandatory 7-digit number @ any domain.<br>Note that the following URI's will match:<br>sip:004412345678@abc.com<br>sip:4412345678@asdf.com<br>sip:12345678@asdf .com<br>sip:2345678@asdf .com<br>sip:012345678@domain.com<br>sip:01512345454@wherever | `sip:(((00)?44)?0?[0-9]{1,3})?([0-9]{7})@(.*)` |

Note that in all of the above expressions, there is no match if the RURI isn't prepended with "`sip:`", i.e. requests beginning with "`tel:`" will not match. Also, while `sip:` is matched, it isn't stored in any of the above examples.

## *Regular Expressions in the Forward To*

The Forward To attribute of the Dial Plan defines where and how to send SIP traffic. A specific destination SIP URI address is defined to forward the call to.

Here you may enter Regular Expressions for the Dial Plan, used to define where and how the firewall should forward the request using the Dial Plan. A Regular Expression refers to Reg Exp subexpressions on the corresponding row in the Matching Request-URI table. Subexpressions are numbered in the order of their starting parenthesis and referred to as $number.

In the expression (sip:(.+))@ingate.com, which matches any Request-URI like sip:user@ingate.com, there are two referable subexpressions: sip:user, which is referred to as $1, and user, which is referred to as $2. You can always refer to the entire Request-URI with $0, as long as the match in the Matching Request-URI table was made using a Reg Exp.

You may define lines in the Dial Plan that lack a Forward to definition. This is useful if you for example are forwarding by ENUM.

| SIP URI Example Description | Equivalent Regular Expression |
|---|---|
| 7-digit number@ Any Domain<br>7-digit number@ IP Address<br>7-digit number@ Domain<br>7-digit number@ (anything hereafter doesn't matter) | `sip:[0-9]{7}@(.*)`<br>`sip:[0-9]{7}@12.34.56.78`<br>`sip:[0-9]{7}@sip_domain.com`<br>`sip:[0-9]{7}@` |
| 7-digit numbers within the London area codes 0207 and 0208 @ | `sip:020[78][0-9]{7}@` |
| 7-digit number with 0845 Lo-call prefix @<br>6 to 7-digit number with 0845 Lo-call prefix @<br>6-digit number with 0845 Lo-call prefix @ | `sip:0845[0-9]{7}@`<br>`sip:0845[0-9]{6,7}@`<br>`sip:0845[0-9]{6}@` |
| 7-digit number with 0870 National prefix @<br>6 to 7-digit number with 0870 National prefix @<br>6-digit number with 0870 National prefix @ | `sip:0870[0-9]{7}@`<br>`sip:0870[0-9]{6,7}@`<br>`sip:0870[0-9]{6}@` |
| 7-digit number with 0845 Lo-call or 0870 National prefix @<br>Note: the first can also match 0875 or 0840, the second wont | `sip:08[74][05][0-9]{7}@`<br>`sip:(0870\|0845)[0-9]{7}@` |
| Local 7-digit number, beginning with optional 9 @ any alphanumeric domain string | `sip:9?[0-9]{7}@(.*)` |
| Use Stored Variable @ IP Address<br>Use Stored Variable @ Domain | `sip:$1@12.34.56.78`<br>`sip:$1@sip_domain.com` |

| | |
|---|---|
| Use Stored Variable @ IP Address with B2BUA<br>Use Stored Variable @ Domain with B2BUA | `sip:$1@12.34.56.78;b2bua`<br>`sip:$1@sip_domain.com;b2bua` |
| Use Stored Variable @ IP Address with Port<br>Use Stored Variable @ Domain with Port | `sip:$1@12.34.56.78:5060`<br>`sip:$1@sip_domain.com:5060` |
| Use Stored Variable @ IP Address with Port and Transport<br>Use Stored Variable @ Domain with Port and Transport | `sip:$1@12.34.56.78:5060;transport=UDP`<br>`sip:$1@sip_domain.com:5060;transport=UDP` |

## *Additional Information*

The Dial-Plan in the Ingate tends to be one of those "set it and forget it" things that one rarely need adjust. So a few tools to help out would be great. Here are a few we recommend to build and test regular expressions before they go live:

echo blah | grep -E (regexp)
http://gskinner.com/RegExr/
google search
http://renschler.net/RegexBuilder/