How To Guide

CHM – Conditional Header Manipulation

2015-03-20

Table of Contents

Tested versions: Ingate Firmware v5.0.4
Ingate Firmware v5.0.5

Revision History:

Revision	Date	Author	Comments
0.1	2013-09-30	Martin Gartner	1st draft
1	2015-03-20	Paul Donald, Martin Gartner	Re-write and clean-up for 1st release for v5.0.4 and v5.0.5. More examples.

This document describes the Conditional Expressions and Regular Expressions (RegExp, RegExr) for GHM or simply Conditional Header Manipulation (CHM) implemented in Ingate firmware 5.0.4 onwards. This document is intended as a supplement to the GHM document available freely on the ingate.com website.

Notes:

Text written in a `Monospace Font` signifies regular expression code i.e. GHM, CHM, CRE, CVS or processing result.

Result samples are not intended to be protocol-accurate or compliant, are intended for demonstration only and reduced for simplicity.

1.1 Conditional Variable Substitution (CVS)

The variable substitution functionality in Ingate firmwares has been enhanced to allow for "conditional variable substitution". It looks like this:

```
$(CONDIF.diversion.user)$(CONDYES.diversion.user)$(CONDNO.from.user)
```

i.e. **IF** there is a `user` portion of a `Diversion:` header present (e.g. from the PBX), then this expression will supply the content of the original `$(diversion.user)`, otherwise it will supply the content of `$(from.user)`

1.2 Conditional Regular Expressions (CRE)

The variable substitution functionality has been enhanced to allow applying regular expressions and replace on the *result* of variable substitutions. For example:

```
$(REGMATCH_123_REGMOD_456_REGELSE_(.*)_REGMOD_789_REGEND.from.user)
```

will first take the `user` portion of the `From:` header, and then perform the regular match/replace which must be enclosed by the tags "`REGMATCH_`" and "`_REGEND`". In the above case;

1. the `user` portion of the `From:` header will be compared with `123`, if the `user` portion matches `123`, then the result would be `456`.
2. otherwise, the `user` portion of the `From:` header will be compared with the regular expression `(.*)` which greedily matches everything, so the result of the above expression would be `789`.

This was a simplified example just to show the basic logic. In practice this can be much more complex.

1.2.1 CRE - Sweden Number Normalization

The following expression will supply number normalization based on the `From:` header of a telephone number in Sweden:

1. If the number already starts with a + followed by only digits, **leave the + and digits as they are**
2. If the number starts with 00 followed by only digits, **replace the 00 with + and leave the digits as they are**
3. if the number starts with 0 followed by only digits, **add +46 followed by the digits after the initial 0**
4. otherwise, **supply the number unmodified**

The following is just one long line:

```
$(REGMATCH_^\+([0-9]+)$_REGMOD_+$1_REGELSE_^00([0-9]+)$_REGMOD_+
$1_REGELSE_^0([0-9]+)
$_REGMOD_+46$1_REGELSE_(.*)_REGMOD_$1_REGEND.from.user)
```

1.3 Together with GHM

The purpose of the PLAIN context is to conditionally include URI-encoded plain-text content. This can be used together with generic header manipulation (GHM), to e.g. populate any given field:

```
?from=$(CONDIF.diversion.user)$(CONDYES.PLAIN.%3csip
%3a12345678%40company.com%3e)$(CONDNO.PLAIN.%3csip%3aABCDEFGH%40company.com
%3e)
```

This does the following:
If there is a `user` portion of a `Diversion:` header present, then the `From:` header will be modified to `<sip:12345678@company.com>`, otherwise it will be modified to `<sip:ABCDEFGH@company.com>`

Resulting in either:

```
...
Diversion: emptyURI
From: <sip:ABCDEFGH@company.com>
```

or

```
...
Diversion: joe@domain
From: <sip:12345678@company.com>
```

1.3.1 More Complex

```
?from=$(CONDIF.diversion.user)$(CONDYES.PLAIN.%3csip%3a)$
(CONDYES.REGMATCH_^\+([0-9]+)$_REGMOD_+$1_REGELSE_^00([0-9]+)$_REGMOD_+
$1_REGELSE_^0([0-9]+)$_REGMOD_+46$1_REGELSE_(.*)_REGMOD_$1_REGEND.from.user)
$(CONDYES.PLAIN.%40company.com%3e)
```

This does the following:

1. If the message which comes from the PBX contains a `Diversion:` header, then the username portion which comes in the `From:` header (appended with "`company.com`") will be put in the `From:` header of the message sent to the service provider, after applying the Sweden number normalization example earlier.
2. If the message which comes from the PBX does not contain a `Diversion:` header, no header manipulation is performed.

Resulting in either an unmodified from header:

```
...
From: <sip:0812345678@company.com>
```

or

```
...
Diversion: mike@domain
From: <sip:+46812345678@company.com>
```

1.3.2 More Complex - History-Info

If the forwarding information is transported via a `History-Info:` header instead, then the following should be the way to go:

```
?from=$(CONDIF.history-info[-1].user)$(CONDYES.PLAIN.%3csip%3a)$
(CONDYES.REGMATCH_^\+([0-9]+)$_REGMOD_+$1_REGELSE_^00([0-9]+)$_REGMOD_+
$1_REGELSE_^0([0-9]+)
$_REGMOD_+46$1_REGELSE_(.*)_REGMOD_$1_REGEND.from.user)$(CONDYES.PLAIN.
%40company.com%3e)
```

Note that `history-info[-1]` refers to the last history-info header present in the SIP message from the PBX.

Resulting in either an unmodified from header:

```
...
From: <sip:0812345678@company.com>
```

or

```
...
History-Info: qwerty
From: <sip:+46812345678@company.com>
```

1.4 URI Parameter Chaining

You can also manipulate multiple headers this way, by chaining the header manipulations in the usual way with the "&" character. The example below is like the previous History-Info example above, plus it sets the field P-Preferred-Identity:

```
?from=$(CONDIF.diversion.user)$(CONDYES.PLAIN.%3csip%3a)$
(CONDYES.REGMATCH_^\+([0-9]+)$_REGMOD_+$1_REGELSE_^00([0-9]+)$_REGMOD_+
$1_REGELSE_^0([0-9]+)$_REGMOD_+46$1_REGELSE_(.*)_REGMOD_$1_REGEND.from.user)
$(CONDYES.PLAIN.%40company.com%3e)&P-Preferred-Identity=%3csip%3aanother_id
%40shop.com%3e
```

Resulting in either an unmodified from header:

```
...
From: <sip:0812345678@company.com>
P-Preferred-Identity: <sip:another_id@shop.com>
```

or

```
...
Diversion: mike@domain
From: <sip:+46812345678@company.com>
P-Preferred-Identity: <sip:another_id@shop.com>
```

Note if you want to add a header (which did not already exist in the message) only under certain circumstances e.g.: Replace a possibly existing `Referred-By:` header with a `Diversion:` header

```
?Diversion=$(CONDIF.Referred-By.user)$(CONDNO.ABORT)$(CONDYES.PLAIN.%3csip
%3a)$(CONDYES.Referred-By.user)$(CONDYES.PLAIN.%40)$(CONDYES.Referred-
By.host)$(CONDYES.PLAIN.%3e)&Referred-By=__remove
```

If no `Referred-By` header is present then without the `$(CONDNO.ABORT)` the following string for header manipulation would remain:

```
?Diversion=
&Referred-By=__remove
```

which would result in an empty diversion header in the resulting message. But with the `$(CONDNO.ABORT)`, the resulting header manipulation string becomes

```
?Diversion=$(ABORT)&Referred-By=__remove
```

and since header manipulations with unresolved variables will be skipped, no empty diversion header will be added. In effect becoming just:

```
?Referred-By=__remove
```

1.5 Illegal Chaining

The following example of chaining is illegal, and will not work:

```
?from=$(CONDIF.diversion.user)$(CONDYES.PLAIN.%3csip%3a&P-
Preferred-Identity=%3csip%3aanother_id%40shop.com%3e)
```

For the above example to compile, it must be corrected – note the closing parenthesis:

```
?from=$(CONDIF.diversion.user)$(CONDYES.PLAIN.%3csip%3a)&P-
Preferred-Identity=%3csip%3aanother_id%40shop.com%3e
```

Summary: Parameter chaining only works outside of the regular expression `$()` context, and cannot be chained conditionally.

2 Keyword / Grammar Summary

2.1.1 Tests

$(

[CONDIF |

CONDIFNOT |

CONDANDIF |

CONDANDIFNOT |

CONDORIF |

CONDORIFNOT].*Header.Portion*

)

2.1.2 Results

$(

CONDYES

.PLAIN.*Text* |

.*Header.Portion* |

.ABORT

| CONDNO

.PLAIN.*Text* |

.*Header.Portion* |

.ABORT

)

2.1.3 RegExp

$(

REGMATCH_

REGMOD

REGELSE

_REGEND

)

NOTE: Keywords and expressions are evaluated from left to right, i.e. forwards in their construction.