# textops Module

**Andrei Pelinescu-Onciul**
**FhG FOKUS**

**Edited by**

**Andrei Pelinescu-Onciul**

**textops Module**

Edited by and Andrei Pelinescu-Onciuland Andrei Pelinescu-Onciul

Copyright © 2003 FhG FOKUS

Revision History

Revision $Revision: 1.1 $ $Date: 2003/07/23 16:20:07 $

# Table of Contents

**1. User's Guide** ..... **1**
1.1. Overview ..... 1
1.1.1. Known Limitations ..... 1
1.2. Dependencies ..... 1
1.2.1. SER Modules ..... 1
1.2.2. External Libraries or Applications ..... 1
1.3. Exported Functions ..... 1
1.3.1. `search(re)` ..... 1
1.3.2. `search_append(re, txt)` ..... 2
1.3.3. `replace(re, txt)` ..... 2
1.3.4. `subst('/re/repl/flags')` ..... 2
1.3.5. `subst_uri('/re/repl/flags')` ..... 3
1.3.6. `append_to_reply(txt)` ..... 3
1.3.7. `append_hf(hf)` ..... 3
1.3.8. `append_urihf(prefix, suffix)` ..... 4
1.3.9. `is_present_hf(hf_name)` ..... 4
1.4. Known Limitatons ..... 4
**2. Developer's Guide** ..... **6**
**3. Frequently Asked Questions** ..... **7**

# List of Examples

1-1. `search` usage ..... 2
1-2. `search_append` usage ..... 2
1-3. `replace` usage ..... 2
1-4. `subst` usage ..... 3
1-5. `subst` usage ..... 3
1-6. `append_to_reply` usage ..... 3
1-7. `append_hf` usage ..... 4
1-8. `append_urihf` usage ..... 4
1-9. `is_present_hf` usage ..... 4

# Chapter 1. User's Guide

## 1.1. Overview

This is mostly an example module. It implements text based operation (search, replace, append a.s.o).

### 1.1.1. Known Limitations

search ignores folded lines. For example, search("(From|f):.*@foo.bar") doesn't match the following From header field:

```
From: medabeda
 <sip:medameda@foo.bar>;tag=1234
```

## 1.2. Dependencies

### 1.2.1. SER Modules

The following modules must be loaded before this module:

- *No dependencies on other SER modules*.

### 1.2.2. External Libraries or Applications

The following libraries or applications must be installed before running SER with this module loaded:

- *None*.

## 1.3. Exported Functions

### 1.3.1. `search(re)`

Searches for the re in the message.

Meaning of the parameters is as follows:

- *re* - Regular expression.

**Example 1-1. `search` usage**

```
...
if ( search("[Ss][Ee][Rr]" ) { /*....*/ };
...
```

### 1.3.2. `search_append(re, txt)`

Searches for the first match of re and appends txt after it.

Meaning of the parameters is as follows:

- *re* - Regular expression.
- *txt* - String to be appended.

**Example 1-2. `search_append` usage**

```
...
search_append("[Ss]er", " blabla");
...
```

### 1.3.3. `replace(re, txt)`

Replaces the first occurence of re with txt.

Meaning of the parameters is as follows:

- *re* - Regular expression.
- *txt* - String.

**Example 1-3. `replace` usage**

```
...
replace("ser", "Sip Express Router");
...
```

### 1.3.4. `subst('/re/repl/flags')`

Replaces re with repl (sed or perl like).

Meaning of the parameters is as follows:

- *'/re/repl/flags'* - sed like regular expression. flags can be a combination of i (case insensitive), g (global) or s (match newline don't treat it as end of line).

**Example 1-4. `subst` usage**

```
...
# replace the uri in to: with the message uri (just an example)
if ( subst('/^To:(.*)sip:[^@]*@[a-zA-Z0-9.]+(.*)$/t:\1\u\2/ig') ) {};
...
```

## 1.3.5. `subst_uri('/re/repl/flags')`

Runs the re substitution on the message uri (like subst but works only on the uri)

Meaning of the parameters is as follows:

- *'/re/repl/flags'* - sed like regular expression. flags can be a combination of i (case insensitive), g (global) or s (match newline don't treat it as end of line).

**Example 1-5. `subst` usage**

```
...
# adds 3463 prefix to numeric uris, and save the original uri (\0 match)
# as a parameter: orig_uri (just an example)
if (subst_uri('/^sip:([0-9]+)@(.*)$/sip:3463\1@\2;orig_uri=\0/i')){$

...
```

## 1.3.6. `append_to_reply(txt)`

Append txt to the reply.

Meaning of the parameters is as follows:

- *txt* - String.

**Example 1-6. `append_to_reply` usage**

```
...
append_to_reply("Foo: bar\r\n");
...
```

## 1.3.7. `append_hf(hf)`

Appends txt after the last header field.

Meaning of the parameters is as follows:

- *hf* - Header field to be appended.

**Example 1-7. `append_hf` usage**

```
...
append_hf("P-hint: VOICEMAIL\r\n");
...
```

## 1.3.8. `append_urihf(prefix, suffix)`

Append header field name with original Request-URI in middle.

Meaning of the parameters is as follows:

- *prefix* - string (usually at least header field name).
- *suffix* - string (usually at least line terminator).

**Example 1-8. `append_urihf` usage**

```
...
append_urihf("CC-Diversion: ", "\r\n");
...
```

## 1.3.9. `is_present_hf(hf_name)`

Return true if a header field is present in message.

> **Note:** Takes header field names "as is" and doesn't distinguish compact names.

Meaning of the parameters is as follows:

- *hf_name* - Header field name.

**Example 1-9. `is_present_hf` usage**

```
...
if (is_present_hf("From")) log(1, "From HF Present");
...
```

## 1.4. Known Limitatons

Search functions are applied to the original request, i.e., they ignore all changes resulting from message processing in SER script.

# Chapter 2. Developer's Guide

The module does not provide any sort of API to use in other SER modules.

# Chapter 3. Frequently Asked Questions

**1.** Where can I find more about SER?

Take a look at http://iptel.org/ser.

**2.** Where can I post a question about this module?

First at all check if your question was already answered on one of our mailing lists:

- http://mail.iptel.org/mailman/listinfo/serusers
- http://mail.iptel.org/mailman/listinfo/serdev

E-mails regarding any stable version should be sent to <serusers@iptel.org> and e-mail regarding development versions or CVS snapshots should be send to <serdev@iptel.org>.

If you want to keep the mail private, send it to <serhelp@iptel.org>.

**3.** How can I report a bug?

Please follow the guidelines provided at: http://iptel.org/ser/bugs