rr Name

Jan Janak
FhG FOKUS

Edited by
Jan Janak

rr Name

Edited by and Jan Janakand Jan Janak

Revision History

Revision $Revision: 1.1 $ $Date: 2003/07/23 16:21:12 $

Table of Contents

List of Examples

Chapter 1. User's Guide

1.1. Overview

The module contains record routing logic

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 Parameters

1.3.1. `enable_full_lr` (integer)

If set to 1 then ;lr=on instead of just ;lr will be used. This is to overcome problems with broken UAs which strip ;lr parameter when generating Route header fields from Record-Route (;lr=on seems to help).

Default value is 0 (no).

Example 1-1. Set `enable_full_lr` parameter

```
...
modparam("rr", "enable_full_lr", 1)
...
```

1.3.2. `append_fromtag` (integer)

if turned on, request's from-tag is appended to record-route; that's useful for understanding whether subsequent requests (such as BYE) come from caller (route's from-tag==BYE's from-tag) or callee (route's from-tag==BYE's to-tag)

Default value is 1 (yes).

Example 1-2. Set `append_fromtag` parameter

```
...
modparam("rr", "append_fromtag", 0)
...
```

1.3.3. `enable_double_rr` (integer)

There are some situations when the server needs to insert two Record-Route header fields instead of one. For example when using two disconnected networks or doing cross-protocol forwarding from UDP->TCP. This parameter enables inserting of 2 Record-Routes. The server will later remove both of them.

Default value is 1 (yes).

Example 1-3. Set `enable_double_rr` parameter

```
...
modparam("rr", "enable_double_rr", 0)
...
```

1.4. Exported Functions

1.4.1. `loose_route()`

The function performs loose routing as defined in RFC3261 (http://www.ietf.org/rfc/rfc3261.txt). See the RFC3261 (http://www.ietf.org/rfc/rfc3261.txt) for more details.

Example 1-4. `loose_route` usage

```
...
loose_route();
...
```

1.4.2. `strict_route()` -- deprecated

If there are any Route HFs in the message, the function takes the first one, rewrites Request-URI with it's value and removes the first URI from Route HFs.

Example 1-5. `strict_route` usage

```
...
strict_route();
...
```

1.4.3. `record_route()`

The function adds a new Record-Route header field. The header field will be inserted in the message before any other Record-Route header fields.

Example 1-6. `record_route` usage

```
...
record_route();
...
```

1.4.4. `record_route_preset(string)`

This function will put the string into Record-Route, don't use unless you know what you are doing.

Meaning of the parameters is as follows:

• *string* - String to be inserted into the header field.

Example 1-7. `record_route_preset` usage

```
...
record_route_preset("1.2.3.4:5090");
...
```

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