

Chameleon LUA for NIT & EPG insertion

Description and background

Insertion of external NIT and “EPG” (EIT tables) in the Chameleon basically includes 2 concepts:

1. Disable the automatically created NIT and EIT tables from the outputs (remux instances)
2. Add the external NIT and EIT.

These operations can be accomplished from the telnet-interface.

Content

1	General table disabling and adding external tables	1
2	Disabling internally created NIT and EIT	2
3	Add the external NIT and EIT.....	2
4	General command for block/un-block of tables	3
4.1	<table_id> values.....	3
5	General PID mapping commands.....	4
5.1	List inputs to find the <input_id>	4
5.2	<input_pid> and <output_pid> for tables	5
6	General instructions for using PuTTY	5

1 General table disabling and adding external tables

The general command for disable (block) or enable a table is:

```
remux.table.set_enable(<remux_key>, <table_id>, <status>)
```

Explanation of the command parameters:

<remux_key>:	Listed by the command =remux.list()
<table_id>:	According to ETSI EN 300 468, see §4.1
<status>:	<i>false</i> to disable, <i>true</i> to enable

The general command for adding a stream from an external source is:

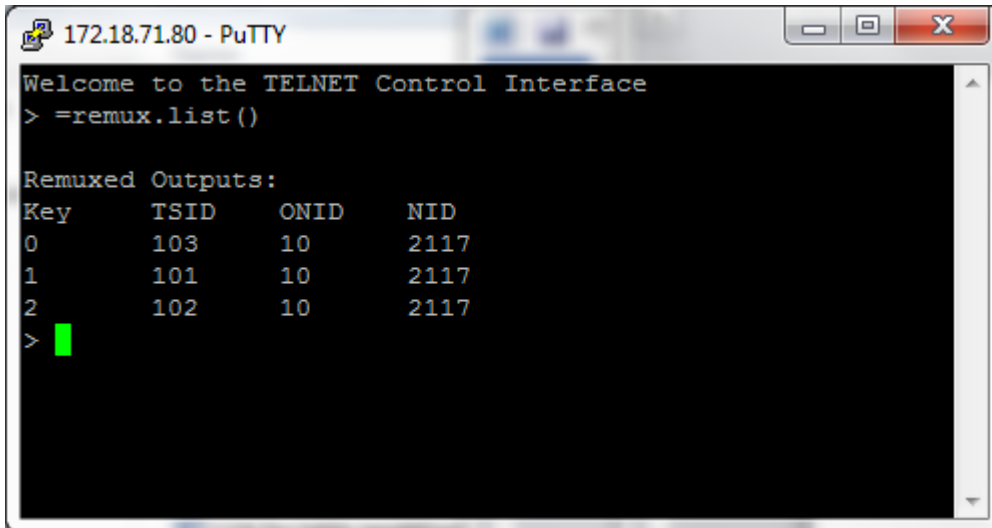
```
remux.pid.user_map(<remux_key>, <input_id>, <input_pid>, <output_pid>)
```

Parameters

<remux_key>:	Reference to the remux instance.
<input_id>:	Input number of stream to map, see §5.1
<input_pid>:	Input pid of stream
<output_pid>:	Output PID to use for stream

2 Disabling internally created NIT and EIT

1. Find the “remux instances”, or “remux keys” by issuing the LUA command:
`=remux.list()`



```
172.18.71.80 - PuTTY
Welcome to the TELNET Control Interface
> =remux.list()

Remuxed Outputs:
Key      TSID    ONID    NID
0        103     10      2117
1        101     10      2117
2        102     10      2117
>
```

2. Disable the internal NIT for each output with the LUA command:
`remux.table.set_enable(<remux_key>, 0x40, false)`
Where <remux key> is the keys in the list from the command =remux.list()
3. Disable the internal EIT for each output with the LUA command:
`for i=0x4E,0x6F do remux.table.set_enable(<remux key>, i, false) end`

Note that there is a set of EIT tables, EIT p/f actual, EIT p/f other (the “now and next”), EIT schedule actual and EIT schedule other, see the table below. Hence the for loop for disabling all these tables. If you would like to disable only a specific EIT table, e.g. the EIT p/f actual, the LUA command is:
`remux.table.set_enable(<remux_key>, 0x4E, false)`

EIT table id values

0x4E	event_information_section - actual_transport_stream, present/following
0x4F	event_information_section - other_transport_stream, present/following
0x50 to 0x5F	event_information_section - actual_transport_stream, schedule
0x60 to 0x6F	event_information_section - other_transport_stream, schedule

3 Add the external NIT and EIT

The general command for adding a stream from an external source is:
`remux.pid.user_map(<remux key>, <input id>, <input pid>, <output pid>)`

[Force passthrough of a stream that is not automatically let through by the remux. Use for streams that is not referenced in any tables.

Parameters

remux_key Reference to the remux instance.
input_id Input number of stream to map
input_pid Input pid of stream
output_pid Output PID to use for stream]

Adding external NIT:

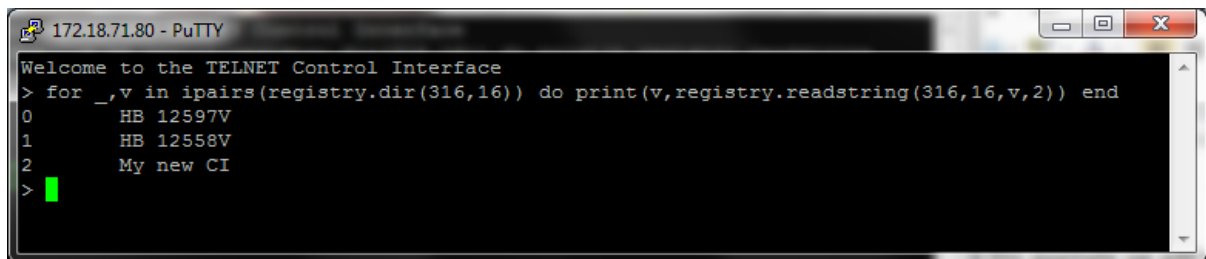
remux.pid.user_map(<remux key>, <input id>, 0x10, 0x10)

The remux id from the command: *=remux.list()*

The input id is listed by the command:

```
for _,v in ipairs(registry.dir(316,16)) do print(v,registry.readstring(316,16,v,2)) end
```

Note: here we assume that the <input pid> is 0x10, as it should be for a NIT. If the NIT you want to insert has another pid for the NIT, please use this as the <input pid>.



Adding external EIT:

remux.pid.user_map(<remux key>, <input id>, 0x12, 0x12)

The remux id from the command: *=remux.list()*

The input id is listed by the command:

```
for _,v in ipairs(registry.dir(316,16)) do print(v,registry.readstring(316,16,v,2)) end
```

Note: here we assume that the <input pid> is 0x12, as it should be for a EIT. If the EIT you want to insert has another pid for the EIT, please use this as the <input pid>.

4 General command for block/un-block of tables

To block automatically created by the Chameleon remux, use the LUA commands:

```
remux.table.set_enable(<remux_key>, <table_id>, false)
```

To un-block a table:

```
remux.table.set_enable(<remux_key>, <table_id>, true)
```

<remux_key> is found using the3 command *=remux.list()*

4.1 <table_id> values

Table 2: Allocation of table_id values

Value	Description
0x00	program_association_section
0x01	conditional_access_section
0x02	program_map_section
0x03	transport_stream_description_section
0x04 to 0x3F	reserved

0x40	network_information_section - actual_network
0x41	network_information_section - other_network
0x42	service_description_section - actual_transport_stream
0x43 to 0x45	reserved for future use
0x46	service_description_section - other_transport_stream
0x47 to 0x49	reserved for future use
0x4A	bouquet_association_section
0x4B to 0x4D	reserved for future use
0x4E	event_information_section - actual_transport_stream, present/following
0x4F	event_information_section - other_transport_stream, present/following
0x50 to 0x5F	event_information_section - actual_transport_stream, schedule
0x60 to 0x6F	event_information_section - other_transport_stream, schedule
0x70	time_date_section
0x71	running_status_section
0x72	stuffing_section
0x73	time_offset_section
0x74	application information section (TS 102 812 [15])
0x75	container section (TS 102 323 [13])
0x76	related content section (TS 102 323 [13])
0x77	content identifier section (TS 102 323 [13])
0x78	MPE-FEC section (EN 301 192 [4])
0x79	resolution notification section (TS 102 323 [13])
0x7A	MPE-IFEC section (TS 102 772 [50])
0x7B to 0x7D	reserved for future use
0x7E	discontinuity_information_section
0x7F	selection_information_section
0x80 to 0xFE	user defined
0xFF	reserved

5 General PID mapping commands

Generally, the PID remapping command for mapping an in-PID to an out-PID is:

```
remux.pid.user_map(<remux key>, <input id>, <input_pid>,<output_pid>)
```

Parameters

<remux_key> Reference to the remux instance, found by the command `=remux.list()`

<input_id> Input number of stream to map, see 4.1

<input_pid> Input pid of stream, see 4.2

<output_pid> Output PID to use for stream, see 4.2

5.1 List inputs to find the <input_id>

```
> for _,v in ipairs(registry.dir(316,16)) do print(v,registry.readstring(316,16,v,2)) end
```

```

0      My DVB-S input
1      My other input stream

```

The command lists all configured inputs. The first column is the input_id. Here, the input_id for “My DVB-S input” is: 0, and the input_id for “My other input stream” is: 1

5.2 <input_pid> and <output_pid> for tables

Table 1: PID allocation for SI

Table	PID value
PAT	0x0000
CAT	0x0001
TSDT	0x0002
reserved	0x0003 to 0x000F
NIT, ST	0x0010
SDT, BAT, ST	0x0011
EIT, ST CIT (TS 102 323 [13])	0x0012
RST, ST	0x0013
TDT, TOT, ST	0x0014
network synchronization	0x0015
RNT (TS 102 323 [13])	0x0016
reserved for future use	0x0017 to 0x001B
inband signalling	0x001C
measurement	0x001D
DIT	0x001E
SIT	0x001F

6 General instructions for using PuTTY

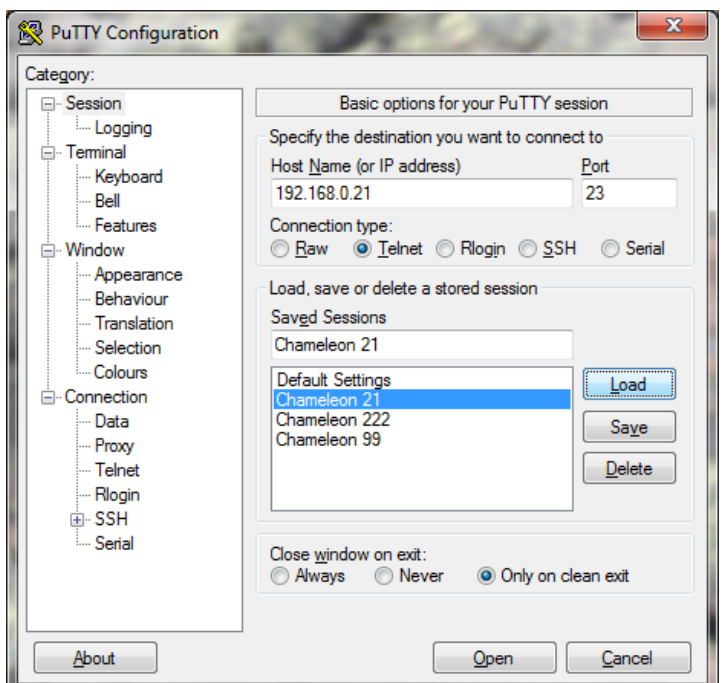
PuTTY download URL: <http://www.putty.org/>

PuTTY is an SSH and telnet client for the Windows platform. PuTTY is open source software.

Some general instructions:

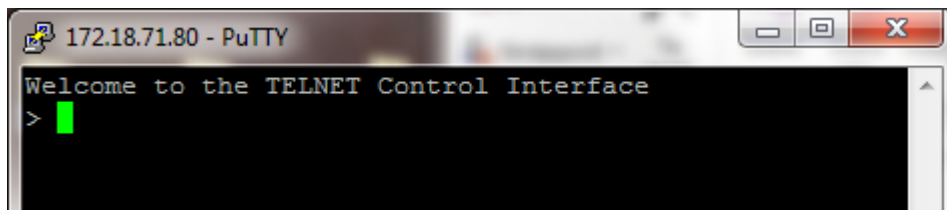
Connect via Putty:

1. Start PuTTY
2. Type the IP address of the Chameleon in the “Host Name (or IP address)” window
3. Enter Port: 23
4. Select connection type: Telnet
5. Click “Open”



As an alternative, you can save a session with the appropriate settings, load it, and possibly only change the IP address

The PuTTY window will appear:



Prompt from Putty: >

If you expect/need a printing in the Putty window from an issued command, put "=" before the LUA command.

Example: to list the services in an output mux with remux key 0, type "=remux.services.list(0)