M3UA Management Commands

Table of Contents

M3UA stack is also responsible to manage the SCTP Associations.

M3UA Management - SCTP

M3UA - SCTP is managed by `sctp` command. It allows to perform following:

- `sctp server create`
- `sctp server destroy`
- `sctp server start`
- `sctp server stop`
- `sctp server show`
- `sctp association create`
- `sctp association destroy`
- `sctp association show`

Create SCTP Server

SCTP Server can be created by issuing command with following structure:

```
sctp server create <server-name> <host-ip> <host-port> <socket-type>
```

Where:

server-name

Unique name assigned to the server.

host-ip

The host ip address where underlying SCTP server socket will bind. For SCTP multi-home support, you can pass comma separated ip addresses that this server socket will bind to. If the primary ip address becomes unavailable, it will automatically fall back to secondary address. For socket-type TCP, comma separated values will be ignored and only first value (primary address) will be used

host-port

The host port where underlying SCTP server socket will bind

socket-type

This is optional. If not passed default is SCTP else specify as TCP.

Example 1. SCTP Server creation

```
mobicents(127.0.0.1:3436)>sctp server create TestServer 127.0.0.1 2905
Successfully added Server=TestServer
```

Example 2. SCTP Server creation with multi-home

```
mobicents(127.0.0.1:3436)>sctp server create TestServer 10.2.50.145,10.2.50.146
2905
Successfully added Server=TestServer
```

Example 3. TCP Server creation

```
mobicents(127.0.0.1:3436)>sctp server create TestServerTCP 10.2.50.145 2906 TCP
Successfully added Server=TestServerTCP
```

Destroy SCTP Server

SCTP Server can be destroyed by issuing command with following structure:

```
sctp server destroy <server-name>
```

Where:

server-name

Unique name of the server to be destroyed. Make sure server is stopped before destroying.

Example 4. Destroy SCTP Server

```
mobicents(127.0.0.1:3436)>sctp server destroy TestServer
Successfully removed Server=TestServer
```

Start SCTP Server

SCTP Server can be started by issuing command with following structure:

```
sctp server start <server-name>
```

Where:

server-name

Unique name of the server to be started. The underlying SCTP server socket is bound to ip:port configured at creation time.

Example 5. Start SCTP Server

```
mobicents(127.0.0.1:3436)>sctp server start TestServer
Successfully started Server=TestServer
```

Stop SCTP Server

SCTP Server can be stopped by issuing command with following structure:

```
sctp server stop <server-name>
```

Where:

server-name

Unique name of the server to be stopped. The underlying socket is closed at this point and all resource are released.

Example 6. Stop SCTP Server

```
mobicents(127.0.0.1:3436)>sctp server stop TestServer
Successfully stopped Server=TestServer
```

Show SCTP Server

SCTP Server's configuration can be viewed by issuing command with following structure:

```
sctp server show
```

Example 7. Show SCTP Server

```
mobicents(local)>sctp server show

SERVER TCP name=TestServerTCP started=false hostIp=10.2.50.145 hostPort=2906
Associations:

SERVER SCTP name=TestServer started=false hostIp=10.2.50.145 hostPort=2905
secondaryHost=10.2.50.146
Associations:
```

Create SCTP Association

Association can be created by issuing command with following structure:

```
sctp association create <assoc-name> <CLIENT | SERVER> <server-name> <peer-ip> <peer-
port> <host-ip> <host-port> <socket-type>
```

Where:

assoc-name

Unique name of the association

CLIENT | SERVER

If this association is client side or server side. If its client side, it will initiate the connection to peer and bind's to host-ip:host-port trying to connect to peer-ip:peer-port. For SCTP multi-home support, you can pass comma separated ip addresses that this association will bind to. If the primary ip address becomes unavailable, it will automatically fall back to secondary address. For socket-type TCP, comma separated values will be ignored and only first value (primary address) will be used

server-name

If this association is server side, server-name must be passed to associate with server. Server with server-name should have already been created by using command Create SCTP Server

socket-type

This is optional. If not passed default is SCTP else specify as TCP. If association is of SERVER type, the socket-type should match with one specified while creating server.

Example 8. Create CLIENT SCTP Association

```
mobicents(local)>sctp association create Assoc1 CLIENT 192.168.56.101 2905
192.168.56.1,192.168.56.1 2905
Successfully added client Associtaion=Assoc1
```

Example 9. Create SERVER SCTP Association

```
mobicents(192.168.56.1:3436)>sctp association create Assoc2 SERVER TestServer
192.168.56.1 2905
Successfully added server Association=TestServer
```

Destroy SCTP Association

Association can be destroyed by issuing command with following structure:

```
sctp association destroy <assoc-name>
```

Where:

assoc-name

Unique name of the association to be destroyed

Example 10. Destroy SCTP Association

```
mobicents(192.168.56.1:3436)>sctp association destroy Assoc1
Successfully removed association=Assoc1
```

Show SCTP Association

Configuration of Association can be viewed by issuing command with following structure:

```
sctp association show
```

Example 11. Show SCTP Association

```
(local)>sctp association show

ASSOCIATION SCTP name=Assoc1 started=false peerIp=192.168.56.101 peerPort=2905
hostIp=192.168.56.1 hostPort2905 type=CLIENT secondaryHost=192.168.56.1

ASSOCIATION SCTP name=Assoc2 started=false peerIp=192.168.56.1 peerPort=2905
server=TestServer type=SERVER
```

M3UA Management

M3UA is managed by `m3ua` command. It allows to perform following:

- `m3ua as create`
- `m3ua as destroy`
- `m3ua as show`
- `m3ua asp create`
- `m3ua asp destroy`
- `m3ua asp show`
- `m3ua asp start`
- `m3ua asp stop`

- `m3ua as add`
- `m3ua as remove`
- `m3ua route add`
- `m3ua route remove`
- `m3ua route show`

Create AS

Application Server (AS) can be created by issuing command with following structure:

```
m3ua as create <as-name> <AS | SGW | IPSP> mode <SE | DE> ipspType <client | server>
rc <routing-context> traffic-mode <traffic mode> min-asp <minimum asp active for
TrafficModeType.Loadshare> network-appearance <network appearance>
```

Where:

as-name

simple string name, which identifies AS. Make sure this is unique. This is mandatory parameter

AS | *SGW* | *IPSP*

Specify if this is of type AS or SGW or IPSP. This is mandatory parameter

SE | *DE*

Specify if the single or double exchange of ASP State Maintenance (ASPSM) and ASP Traffic Maintenance (ASPTM) messages should be performed. This is mandatory parameter.

client | *server*

If As if of type IPSP, speicfy here if its client or server type.

routing-context

refers to Routing Context already configured on M3UA stack on SGW side. This is optional parameter. If no Routing Context is passed, Application Server Process (assigned to this AS) may not be configured to process signaling traffic related to more than one Application Server, over a single SCTP Association

traffic-mode

Traffic mode for ASP's. By default its loadshare. Mobicents M3UA only supports loadshare and override, broadcast is not supported.

min-asp

The minumum number of active ASPs needed (if the traffic mode is 'loadshare') before the payload starts flowing (AS goes into ACTIVE state). This is an optional parameter and if not specified the default value is 1. Also if the traffic-mode is not set as 'loadshare', this parameter has no effect.

network-appearance

The Network Appearance is a M3UA local reference shared by SG and AS (typically an integer)

that, together with an Signaling Point Code, uniquely identifies an SS7 node by indicating the specific SS7 network to which it belongs. It can be used to distinguish between signaling traffic associated with different networks being sent between the SG and the ASP over a common SCTP association. This is optional.

Example 12. AS (IPSP) creation

```
mobicents(127.0.0.1:3435)>m3ua as create AS1 IPSP mode DE ipspType server rc 1
traffic-mode loadshare
Successfully created AS name=AS1
```

Example 13. AS creation

```
mobicents(127.0.0.1:3435)>m3ua as create AS2 AS mode SE rc 100 traffic-mode
loadshare network-appearance 12
Successfully created AS name=AS2
```

Destroy AS

Application Server (AS) can be destroyed by issuing command with following structure:

```
m3ua as destroy <as-name>
```

Where:

as-name

Simple string name, which identifies AS. Make sure AS is in state INACTIVE and all the ASP's are unassigned before destroying

Example 14. Destroy AS

```
mobicents(127.0.0.1:3435)>m3ua as destroy AS1
Successfully destroyed AS name=AS1
```

Show AS

Application Server configured can viewed by issuing command with following structure:

```
m3ua as show
```

Example 15. Show AS

```
mobicents(local)>m3ua as show

AS name=AS2 functionality=AS mode=SE rc=[100] trMode=2 defaultTrMode=2 na=12
peerFSMState=DOWN
Assigned to :


AS name=AS1 functionality=IPSP mode=DE ipspType=SERVER rc=[1] trMode=2
defaultTrMode=2 localFSMState=DOWN peerFSMState=DOWN
Assigned to :
```

Create ASP

Application Server Process (ASP) can be created by issuing command with following structure:

```
m3ua asp create <asp-name> <sctp-association> aspid <aspid>
```

Where:

asp-name
: Name of this ASP. It should be unique

sctp-association
: name of SCTP Association

aspid
: Identifier for this newly created Application Server Process. If this is not passed, next available aspid will be used.

Example 16. ASP creation

```
mobicents(127.0.0.1:3435)>m3ua asp create ASP1 Assoc1
Successfully created AS name=ASP1
```

Destroy ASP

ASP can be destroyed by issuing command with following structure:

```
m3ua asp destroy <asp-name>
```

Where:

asp-name
: Name of this ASP to be destroyed. Make sure ASP is stopped before destroying

Example 17. Destroy ASP

```
mobicents(127.0.0.1:3435)>m3ua asp destroy ASP1
Successfully destroyed ASP name=ASP1
```

Show ASP

ASP configured can be viewed by issuing command with following structure:

```
m3ua asp show
```

Example 18. Show ASP

```
mobicents(local)>m3ua asp show

ASP name=ASP1 sctpAssoc=Assoc1 started=false
Assigned to :
```

Start ASP

Application Server Process (ASP) can be started with following structure

```
m3ua asp start <asp-name>
```

Where:

asp name
: name of ASP created earlier. Make sure ASP you are trying to start is assigned to at least one AS

Example 19. Start ASP

```
mobicents(127.0.0.1:3435)>m3ua asp start ASP1
Successfully started ASP name=ASP1
```

Stop ASP

Application Server Process (ASP) can be stopped with following structure

```
m3ua asp stop <asp-name>
```

Where:

asp name
: name of ASP started earlier.

Example 20. Stop ASP

```
mobicents(127.0.0.1:3435)>m3ua asp stop ASP1
Successfully stopped ASP name=ASP1
```

Add ASP to AS

Application Server Process (ASP) can be assigned to Application Server (AS) with following structure

```
m3ua as add <as-name> <asp-name>
```

Where:

as name
: name of AS created earlier

asp name
: name of ASP created earlier

Note

Mobicents M3UA supports configuring ASP to process signaling traffic related to more than one Application Server, over a single SCTP Association. However you need to make sure that all the AS's that ASP is shared with has Routing Context (unique) configured.

Example 21. Add ASP to AS

```
mobicents(127.0.0.1:3435)>m3ua as add AS1 ASP1
Successfully added ASP name=ASP1 to AS name=AS1
```

Remove ASP from AS

Application Server Process (ASP) can be unassigned from Application Server (AS) with following structure

```
m3ua as remove <as-name> <asp-name>
```

Where:

as name
: name of AS

asp name
: name of ASP

Example 22. Remove ASP from AS

```
mobicents(127.0.0.1:3435)>m3ua as remove AS1 ASP1
Successfully removed ASP name=ASP1 from AS name=AS1
```

Add Route

Configure the destination point code that message will be routed to

```
m3ua route add <as-name> <dpc> <opc> <si>
```

Where:

as name
: name of AS created earlier

dpc
: Destination point code

opc
: Originating point code

si
: Service Indicator

Example 23. Add Route

```
mobicents(127.0.0.1:3435)>m3ua route add AS1 2 -1 -1
```

Remove Route

Remove the As configured for the destination point code

```
m3ua route remove <as-name> <dpc> <opc> <si>
```

Where:

as name
: name of AS assigned to route message for this dpc

dpc
: Destination point code

opc
: Originating point code

si
: Service Indicator

Example 24. Remove Route

```
mobicents(127.0.0.1:3435)>m3ua route remove AS1 2 -1 -1
```

Show Route

Show all the routes configured

```
m3ua route show
```

Example 25. Show Route

```
mobicents(local)>m3ua route show

2:-1:-1        AS1,AS2,
```