Table of Contents

[1. Administration 2](#_Toc453328478)

[1.1 Domains, machines and processes 2](#_Toc453328479)

[1.2 Directories 2](#_Toc453328480)

[1.3 Log files 2](#_Toc453328481)

[1.4 Management 3](#_Toc453328482)

[1.4.1 Starting and stopping processes 3](#_Toc453328483)

[1.4.2 Checking IM-SCF components 5](#_Toc453328484)

[1.4.3 Managing SCTP links 9](#_Toc453328485)

[1.5 Configuration 10](#_Toc453328486)

[1.5.1 Editing XML files manually 10](#_Toc453328487)

[1.5.2 Configuring logging 11](#_Toc453328488)

[1.5.3 IM-SCF configuration reference 13](#_Toc453328489)

[1.5.4 Using IM-SCF configuration scripts 26](#_Toc453328490)

[1.5.5 Deploying new IM-SCF binaries 38](#_Toc453328491)

[1.6 Backup and restore 39](#_Toc453328492)

[1.6.1 Create a backup 39](#_Toc453328493)

[1.6.2 Uploading backups 40](#_Toc453328494)

[1.6.3 Restoring from backup 40](#_Toc453328495)

# Administration

## Domains, machines and processes

IM-SCF deployments are managed in terms of domains. A domain is an administrative concept of managed servers configured for the same purpose. A domain has two types of servers (instances):

* Signaling Layer (SL) servers act as a communication layer with the telco network over SIGTRAN
* Execution Layer (EL) servers are the “brain” – they process the low-level telco messages from SL, communicate with the AS layer over SIP and construct the messages to be sent back to the telco network (and send them through SL)

Each server is a WildFly application server instance. The IM-SCF is basically an enterprise Java application deployed into this WildFly application server. We use the WildFly instances in a so-called standalone mode, so even instances in the same IM-SCF domain do not communicate with each other on application server level.

There can be at most two SL servers per domain. Since an SL server’s task is simple, they are lightweight servers and the two-server limit is not holding back throughput. Scaling is done by installing more EL servers in the domain since the processing is done in the EL servers.

Configuration of a domain is stored in an XML file which can be edited by hand as well but the IM-SCF package provides quick scripts for the most common everyday tasks to minimize human errors.

## Directories

| **Directory** | **Description** |
| --- | --- |
| /home/users/imscfadmin | imscfadmin user home directory |
| /home/users/monitor | monitor user home directory |
| /home/imscfadmin | Symbolic link to /home/users/imscf™admin |
| /usr/imscf/imscf\_1\_0 | IM-SCF installation |
| /usr/imscf/jdk\_xxx | Installed Java runtime |
| /usr/imscf/java | Symlink to the currently used Java runtime |
| /usr/imscf/trace | Directory of all log files separated by instance |
| /usr/imscf/servers | IM-SCF instance runtime files (configuration, deployment and temporary |
| /home/users/imscfadmin/startup | Start/stop and configuration scripts |
| /home/users/imscfadmin/trace | Symlink to /usr/imscf/trace |

## Log files

All log files are located in directory

/usr/imscf/trace

The symbolic link

/home/users/imscfadmin/trace

points to the directory /usr/imscf/trace.

All instance’s logs are located in a separate directory under /usr/imscf/trace. The name of the directory is the instance name.

In case of managed servers, inside the instance’s directory there are the following files:

| **Log file** | **Description** |
| --- | --- |
| <instance>.out | Standard error and standard output of the process |
| <instance>.gclog | The Java Garbage Collector log of the process |
| <instance>.log | WildFly server log |
| imscf.log | IM-SCF technical log |
| imscf.audit.log | IM-SCF audit log (one line per call) |
| archive/ | Historical .out, .gclog, .log, imscf.log and imscf.audit.log files are kept here |
| logrotate.conf | Log rotation configuration (do not edit, created by script) |
| logrotate.status | The result of the last logrotate operation |

## Management

Management and configuration is done by shell scripts shipped with IM-SCF. These scripts reside on all machines, but their master version as to be on a dedicated operation & managemnt machine that is referred as “oam01” machine in this documentation. If scripts change, they are modified on the oam01 machine and distributed to all other machines by invoking syncScripts.sh. The syncScripts.sh does not have any parameter.

### Starting and stopping processes

#### Starting and stopping instances

To start an IM-SCF instance on a machine use the startIMSCFServerInstance.sh script:

$ startIMSCFServerInstance.sh <domain with type> <instance>

where

* <domain with type> is the name of domain together with the instance type (“SL” or “EL”)
* <instance> is the leg index (“01” or “02”) and the instance index (“a”, “b” or in case of sful domain, “c”)

If you run the script without arguments, it will write out the instances deployed on the current machine:

$ startIMSCFServerInstance.sh   
Setting IM-SCF Environment  
This is a LIVE node on site1-mobile05  
**SL server: slessSL 01a  
EL servers: slessEL 01a 01b**

ERROR: Too few parameters.  
Usage:   
/home/imscfadmin/startup/startstop/startIMSCFServerInstance.sh [-h|--help] domainWithType server

Starts an IM-SCF server instance of a domain.  
Where domainWithType is e.g. gyorSL/slessEL/sfulSL etc, server is the target instance e.g. 01a,01b,02a,02b.

For stopping an IM-SCF instance, use the stopIMSCFServcerInstance.sh script:

$ stopIMSCFServerInstance [-k] <domain with type> <instance>

The parameters <domain with type> and <instance> mean the same as above. If the –k switch is specified on the command line the script will issue a “kill” command with the PID of the process, so the shutdown is rapid. Use the –k switch when shutting down all the instances in a domain.

#### Starting and stopping multiple servers

**Complete system startup**

$ startAll.sh

Use this command when you want all servers up and running on the current machine. The script first checks for all instances deployed on the current machine prior to trying to start them so tries to start only the instances which are not running yet.

**Complete system shutdown**

$ stopAll.sh [-k]

This command stops all IM-SCF instances running on the machine it is issued on. If the –k switch is specified on the command line the script will issue a “kill” command with the PID of the process, so the shutdown is rapid.

**Start specific servers of a domain**

$ startIMSCFServers.sh <domain with type> <instance1> [<instance2>…]

For example:

$ startIMSCFServers.sh sfulEL 02a 02b

Using this command you can start instances of a domain sequentially. Specify the domain in the <domain with type> parameter and add one or more servers as the <instance> parameter to the command line. The construction of the parameters is exactly like above.

$ stopIMSCFServers.sh [-k] <domain with type> <instance1> [<instance2>…]

For example the command:

$ stopIMSCFServers.sh sfulEL 01a 01b 01c

Sequentially stops all the EL servers in the sful domain on the current machine. The parameters <domain with type> and <instance> should be given exactly as above. If the –k switch is specified on the command line the script will issue a “kill” command with the PID of the process, so the shutdown is rapid.

#### Resetting all servers

There is a script which re-sets all instances on the current machine:

$ resetAll.sh

It does not have arguments, since the discovery of the instances deployed on the current machine is automatic. The script does the following:

* Terminates all running IM-SCF instances
* Resets the IM-SCF instances from WildFly AS point of view: they will be now just “bare” WildFly instances
* Starts the IM-SCF instances, now they start but have IM-SCF undeployed
* Deploys IM-SCF binaries to the instances

This script is advised to use after a new IM-SCF version is shipped or a transient error occurred which left at least one of the servers in an unstable state (e.g. disk full).

### Checking IM-SCF components

#### Checking IM-SCF processes and SCCP connections

You can use the checkAll.sh script to check which IM-SCF processes are running on the current machine. The script is accessible from anywhere when logged in as imscfadmin user.

The script pointCodeChecker.sh is used to check one SL instance.

The output of the checkAll.sh script is like the following:

$ checkAll.sh

Setting IM-SCF Environment

This is a PILOT node on pilot-mobile03

--------------------------------------[EL Servers and Deployment Statuses]--------------------------------------

Checking the servers and deployments on domain slessEL with servers: 01a 01b

Checking server 01a on domain slessEL

slessEL01a status is: running

[OK]

Status of deployments:

NAME RUNTIME-NAME PERSISTENT ENABLED STATUS

imscf-el.war imscf-el.war true true OK

[OK]

Checking server 01b on domain slessEL

slessEL01b status is: running

[OK]

Status of deployments:

NAME RUNTIME-NAME PERSISTENT ENABLED STATUS

imscf-el.war imscf-el.war true true OK

[OK]

----------------------------------------------------------------------------------------------------------------

--------------------------------------[SL Servers and Deployment Statuses]--------------------------------------

Checking the servers and deployments on domain slessSL with servers: 01a

Checking server 01a on domain slessSL

slessSL01a status is: running

[OK]

Status of deployments:

NAME RUNTIME-NAME PERSISTENT ENABLED STATUS

imscf-sl.war imscf-sl.war true true OK

[OK]

----------------------------------------------------------------------------------------------------------------

--------------------------------------[ PointCode Statuses ]--------------------------------------

Checking the configured pointcodes on slessSL01a...

Connecting to 192.168.2.21:11111

PointCodes and Statuses:

Name=HLRFE0, PointCode=216, Status=ACTIVE

Name=HLRFE1, PointCode=202, Status=ACTIVE

Name=HLRFE2, PointCode=203, Status=ACTIVE

Name=MSB2, PointCode=272, Status=ACTIVE

Name=MSS0, PointCode=280, Status=ACTIVE

Name=MSS1, PointCode=281, Status=ACTIVE

Name=MSS2, PointCode=282, Status=ACTIVE

Name=MSS3, PointCode=283, Status=ACTIVE

Name=MSS4, PointCode=284, Status=ACTIVE

Name=MSS5, PointCode=285, Status=ACTIVE

Name=MSS6, PointCode=286, Status=ACTIVE

Closing JMX connection...

Done.

----------------------------------------------------------------------------------------------------------------

--------------------------------------[ SCTP associations ]--------------------------------------

Connecting to 192.168.2.21:11111

SCTP associations:

RS-CDF1 - [STARTED][UP]

RS-CDF2 - [STARTED][UP]

RS-HLRFE0 - [STARTED][UP]

RS-HLRFE1 - [STARTED][UP]

RS-HLRFE2 - [STARTED][UP]

RS-MSB2A - [STARTED][UP]

RS-MSB2B - [STARTED][UP]

RS-MSS0A - [STARTED][UP]

RS-MSS0B - [STARTED][UP]

RS-MSS1A - [STARTED][UP]

RS-MSS1B - [STARTED][UP]

RS-MSS2A - [STARTED][UP]

RS-MSS2B - [STARTED][UP]

RS-MSS3A - [STARTED][UP]

RS-MSS3B - [STARTED][UP]

RS-MSS4A - [STARTED][UP]

RS-MSS4B - [STARTED][UP]

RS-MSS5A - [STARTED][UP]

RS-MSS5B - [STARTED][UP]

RS-MSS6A - [STARTED][UP]

RS-MSS6B - [STARTED][UP]

Closing JMX connection...

Done.

----------------------------------------------------------------------------------------------------------------

--------------------------------------[ SIP Access Points ]--------------------------------------

Checking the configured SIP AS Access points on slessEL...

Sip Application Server Access Points and Statuses

+-----------------------------------+---------------+------+-------------------+-------------+-------------+

| imscf5a\_calltester\_all instances | IP | Port | Heartbeat Enabled | slessEL01a | slessEL01b |

+-----------------------------------+---------------+------+-------------------+-------------+-------------+

| calltester\_all-1 | 192.168.1.111 | 8115 | true | UNREACHABLE | REACHABLE |

| calltester\_all-2 | 192.168.1.112 | 8115 | true | UNREACHABLE | REACHABLE |

+-----------------------------------+---------------+------+-------------------+-------------+-------------+

+-----------------------------------+---------------+------+-------------------+-------------+-------------+

| imscf5a\_calltester\_camel instances | IP | Port | Heartbeat Enabled | slessEL01a | slessEL01b |

+-----------------------------------+---------------+------+-------------------+-------------+-------------+

| calltester\_camel-1 | 192.168.1.121 | 8125 | true | REACHABLE | REACHABLE |

| calltester\_camel-2 | 192.168.1.121 | 8125 | true | REACHABLE | REACHABLE |

+-----------------------------------+---------------+------+-------------------+-------------+-------------+

+-----------------------------------+---------------+------+-------------------+-------------+-------------+

| imscf5c\_odomino\_camel instances | IP | Port | Heartbeat Enabled | slessEL01a | slessEL01b |

+-----------------------------------+---------------+------+-------------------+-------------+-------------+

| odomino\_camel-1 | 192.168.1.121 | 8314 | true | REACHABLE | REACHABLE |

+-----------------------------------+---------------+------+-------------------+-------------+-------------+

Closing JMX connection...

Done.

Closing JMX connection...

Done.

----------------------------------------------------------------------------------------------------------------

--------------------------------------[ CHECKALL SUMMARY ]--------------------------------------

Statuses of the EL servers and the deployments...

[OK]

Statuses of the SL servers and the deployments...

[OK]

Statuses of the PointCodes...

[OK]

SCTP associations and statuses...

[OK]

Statuses of the SIP AS access points...

[OK]

----------------------------------------------------------------------------------------------------------------

The pointCodeChecker.sh is located in directory /home/imscfadmin/startup/tools and is not on the path so must be run either from there or the relative path specified. Its parameters is the domain with server type (e.g. slessSL since it can be called for Signaling Layer servers) and the instance name (e.g. 01a). The output can be like:

imscfadmin@site1-mobile03:~/startup/tools$ ./pointCodeChecker.sh slessSL 01a

Setting IM-SCF Environment

This is a LIVE node on site1-mobile03

SL server: slessSL 01a

EL servers: slessEL 01a 01b

Checking the configured pointcodes on perfSL01a...

Connecting to 192.168.2.21:11111

PointCodes and Statuses:

Name=MSS0, PointCode=280, Status=ACTIVE

Name=MSS1, PointCode=281, Status=ACTIVE

Name=MSS2, PointCode=282, Status=ACTIVE

Name=MSS3, PointCode=283, Status=ACTIVE

Name=MSS4, PointCode=284, Status=ACTIVE

Name=MSS5, PointCode=285, Status=ACTIVE

Name=MSS6, PointCode=286, Status=ACTIVE

Closing JMX connection...

Done.

#### Checking HLR query functionality

You can check if IM-SCF HLR query subsystem is up and working by invoking the startup / atifnr / checkHlrFnr.sh script. The scripts check if ATI and FNR queries are working for all configured HLR instances on all IM-SCF endpoints. The script must be invoked on the oam01 machine and the output is similar to the following if everything is okay:

192.168.1.51 fnr HLRFE0FNR Target instance=1a

SUCCESS

192.168.1.51 ati HLRFE0 Target instance=1a

SUCCESS

192.168.1.51 fnr HLRFE0FNR Target instance=1b

SUCCESS

192.168.1.51 ati HLRFE0 Target instance=1b

SUCCESS

192.168.1.51 fnr HLRFE1FNR Target instance=1a

SUCCESS

192.168.1.51 ati HLRFE1 Target instance=1a

SUCCESS

192.168.1.51 fnr HLRFE1FNR Target instance=1b

SUCCESS

192.168.1.51 ati HLRFE1 Target instance=1b

SUCCESS

192.168.1.51 fnr HLRFE2FNR Target instance=1a

SUCCESS

192.168.1.51 ati HLRFE2 Target instance=1a

SUCCESS

192.168.1.51 fnr HLRFE2FNR Target instance=1b

SUCCESS

192.168.1.51 ati HLRFE2 Target instance=1b

SUCCESS

192.168.1.52 fnr HLRFE0FNR Target instance=2a

SUCCESS

192.168.1.52 ati HLRFE0 Target instance=2a

SUCCESS

192.168.1.52 fnr HLRFE0FNR Target instance=2b

SUCCESS

192.168.1.52 ati HLRFE0 Target instance=2b

SUCCESS

192.168.1.52 fnr HLRFE1FNR Target instance=2a

SUCCESS

192.168.1.52 ati HLRFE1 Target instance=2a

SUCCESS

192.168.1.52 fnr HLRFE1FNR Target instance=2b

SUCCESS

192.168.1.52 ati HLRFE1 Target instance=2b

SUCCESS

192.168.1.52 fnr HLRFE2FNR Target instance=2a

SUCCESS

192.168.1.52 ati HLRFE2 Target instance=2a

SUCCESS

192.168.1.52 fnr HLRFE2FNR Target instance=2b

SUCCESS

192.168.1.52 ati HLRFE2 Target instance=2b

SUCCESS

### Managing SCTP links

It is possible to query the status of and manage the SCTP associations on a specific SL server. The /home/imscfadmin/tools/linkManager.sh does this job. You can query the status of all defined links on a server with the --list switch:

imscfadmin@site1-mobile03:~/startup$ tools/**linkManager.sh slessSL 01a --list**

Setting IM-SCF Environment

This is a LIVE node on site1-mobile03

SL server: slessSL 01a

EL servers: slessEL 01a 01b

Connecting to 192.168.2.21:13111

SCTP associations:

RS-MSS0A - [STARTED][UP]

RS-MSS0B - [STARTED][UP]

RS-MSS1A - [STARTED][UP]

RS-MSS1B - [STARTED][UP]

RS-MSS2A - [STARTED][UP]

RS-MSS2B - [STARTED][UP]

RS-MSS3A - [STARTED][UP]

RS-MSS3B - [STARTED][UP]

RS-MSS4A - [STARTED][UP]

RS-MSS4B - [STARTED][UP]

RS-MSS5A - [STARTED][UP]

RS-MSS5B - [STARTED][UP]

RS-MSS6A - [STARTED][UP]

RS-MSS6B - [STARTED][UP]

Closing JMX connection...

Done.

Associations can be stopped and started with the switches --stop and --start respectively. Note that only M3UA associations can be started or stopped. Managing SCTP associations for Diameter connections is not supported.

Note that when you stop an association which has generated an alert because it is down, the alert will be ceased when the association is stopped. This is because IM-SCF does not consider a link erroneous if it is down and stopped intentionally.

$ **tools/linkManager.sh perfSL 01a --stop RS-MSS0A**

Setting IM-SCF Environment

This is a LIVE node on site1-mobile03

SL server: slessSL 01a

EL servers: slessEL 01a 01b

Connecting to 192.168.2.21:13111

Stopping association=RS-MSS0A

Closing JMX connection...

Done.

$ **tools/linkManager.sh perfSL 01a --list**

Setting IM-SCF Environment

This is a LIVE node on site1-mobile03

SL server: slessSL 01a

EL servers: slessEL 01a 01b

Connecting to 192.168.2.21:13111

SCTP associations:

RS-MSS0A - [STOPPED][DOWN]

RS-MSS0B - [STARTED][UP]

(…)

Closing JMX connection...

Done.

$ **tools/linkManager.sh perfSL 01a --start RS-MSS0A**

Setting IM-SCF Environment

This is a LIVE node on site1-mobile03

SL server: slessSL 01a

EL servers: slessEL 01a 01b

Connecting to 192.168.2.21:13111

Starting assoctiation=RS-MSS0A

Closing JMX connection...

Done.

$ **tools/linkManager.sh perfSL 01a --list**

Setting IM-SCF Environment

This is a LIVE node on site1-mobile03

SL server: slessSL 01a

EL servers: slessEL 01a 01b

Connecting to 192.168.2.21:13111

SCTP associations:

RS-MSS0A - [STARTED][UP]

RS-MSS0B - [STARTED][UP]

(…)

## Configuration

This section explains how to configure IM-SCF domains. Since IM-SCF domain configuration is stored in a human-readable XML file, all configuration can be done by editing this XML file. However, manually editing this XML file is prone to human errors so helper scripts are provided for the everyday tasks.

### Editing XML files manually

There is a single XML configuration file for each domain in the oam01 machine. The file can be found at location

/usr/imscf/configuration/<domain>/<domain>\_imscf\_config.xml

This file contains all information needed to set up the domain’s SL and EL instances.

After you change the XML files either by hand, you should distribute the configuration to the nodes. Use the distributeConfig.sh script for that purpose:

distributeConfig.sh [-l|--reloadLogConfig] [-a|--reloadIMSCFConfig] [-b|reloadBothConfigs] <domain>

The script copies the log and IM-SCF configuration to the machines belonging to the domain specified by the parameter <domain>. You can optionally instruct the script to reload the logging (-l) or IM-SCF (-a) or both (-b) configurations on the domain’s running instances.

**Important note:** When specifying “–a” or “–b” to instruct the domain’s servers to reload their configuration, this configuration reload does not affect SIGTRAN and DIAMETER settings of the Signaling Layer servers. After changing these settings, the affected Signaling Layer servers must be restarted for the changes to take effect.

### Configuring logging

#### Configuring logging using scripts

You can use the setLogLevels.sh script on the OAM machine to set the log levels on the desired domains.

$ setLogLevels.sh   
Setting IM-SCF Environment  
This is a PILOT node on pilot-oam01

ERROR: domain must be specified  
Usage:

/home/imscfadmin/startup/logconfig/setLogLevels.sh [-d DOMAIN|--domain DOMAIN] [-t TYPE|--type TYPE] [-l LEVEL|--level LEVEL] [-s LEG|--side LEG] [-h|--help]

Sets the current loglevel given by the level parameter of the given domain and type. The leg parameter is mandatory, if the domain is used in split mode!  
Where the domain is the name of the domain without the SL/EL postfixes, e.g. gyor, sless, sful, and the type is the type of the domain (SL or EL).   
The acceptable types are only SL and EL (the script is case insensitive in this regard). The acceptable loglevels are: TRACE, DEBUG, INFO, WARN, ERROR.  
This parameter is also case insensitive.

In most cases, the –d (domain), -t (type) and –l (level) parameters are used, so for example for setting the log level of EL servers in sful domain to WARN, issue:

$ setLogLevels.sh –d sful –t EL –l WARN

After the logging configuration files are changed, invoke the distributeConfig.sh command as imscfadmin:

$ distributeConfig.sh [-l] <domain>

The <domain> parameter specifies the target domain where the configuration change has been done. If you specify the optional –l (that is small letter “L”) flag, the logging configuration will be reloaded runtime in the running instances of the domain.

**Important: as seen in Editing XML files manually the distributeConfig.sh script copies the IM-SCF configuration files together with logging configuration so if you have modified the IM-SCF configuration, be prepared that this script copies it to the target machines!**

It is possible to query the current logging settings with the script showLogLevels.sh. The two required parameters are the domain (e.g. “sless”) and the server type (“SL” or “EL”). For example:

$ showLogLevels.sh sful EL  
Setting IM-SCF Environment  
This is a PILOT node on pilot-oam01  
The current loglevel of the sful domain with the type of EL is: WARN

#### Configuring logging using logback XML files

IM-SCF instances main log is written into file imscf.log, see section Log files. The layout and the level of logging can be configured separately for SL and EL instances in XML files. The XML files for SL and EL instances are in files

/usr/imscf/configuration/<domain>/<domain>\_sl\_logback.xml  
/usr/imscf/configuration/<domain>/<domain>\_el\_logback.xml

respectively.

IM-SCF logging level in the shipped configuration is set to level WARN which logs only suspicious or erroneous situations. This is usually sufficient but there can be situations when the operational team wants to see more detailed information about the behavior of IM-SCF, in this case the level has to be set lower.

To change the log level, modify the files at the part below:

<root level="**WARN**">  
 <appender-ref ref="APPLOG"/>  
</root>

The options to change the “WARN” above are following:

* INFO: IM-SCF startup/shutdown events are logged as well
* DEBUG: Call handling information events are logged as well. This usually includes main decision points. This level is usually producing hundreds of lines per call
* TRACE: Logs every event during a call handling. This level can produce thousand lines per call

After the logging configuration files are changed, invoke the distributeConfig.sh command as imscfadmin:

$ distributeConfig.sh [-l] <domain>

The <domain> parameter specifies the target domain where the configuration change has been done. If you specify the optional –l (that is small letter “L”) flag, the logging configuration will be reloaded runtime in the running instances of the domain.

**Important: as seen in section “Editing XML files manually” the distributeConfig.sh script copies the IM-SCF configuration files together with logging configuration so if you have modified the IM-SCF configuration, be prepared that this script copies it to the target machines!**

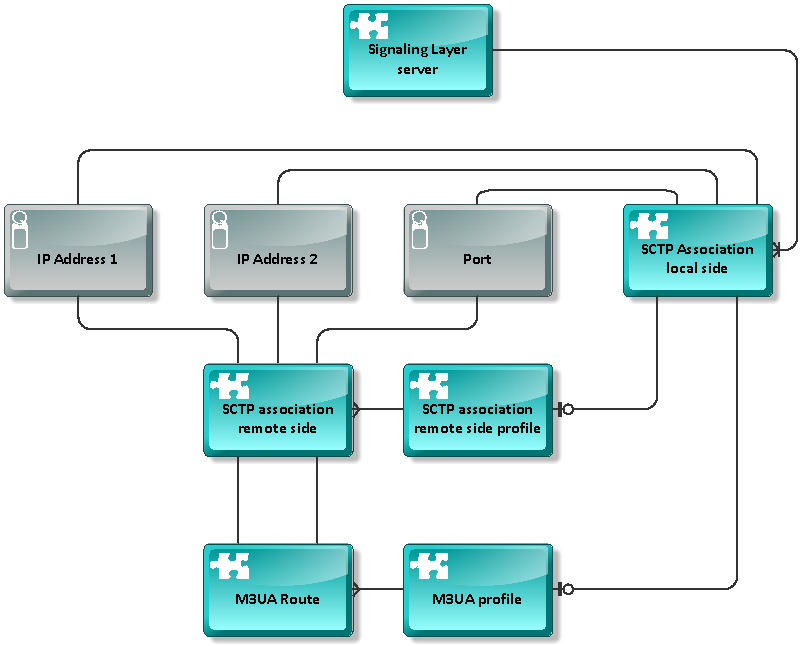
### IM-SCF configuration reference

The parts below explain the structure of the IM-SCF configuration XML.

#### SIGTRAN configuration

##### SCTP and M3UA

For a Signaling Layer server’s SCTP and M3UA configuration, profiles are used. The overview of the entities is shown on the following figure.



The configuration XML starts with defining SCTP association remote sides:

<?xml version=**"1.0"** encoding=**"UTF-8"**?>

<imscfConfig version=**"1.0"** xmlns=**"http://common.imscf.restcomm.org/config"**>

<sctpAssociationRemoteSides>

<sctpAssociationRemoteSide name=**"RS-MSS0A"**>

<remoteSystemType>**SG**</remoteSystemType>

<associationType>**m3ua**</associationType>

<remoteIp1>**10.133.128.2**</remoteIp1>

<remoteIp2>**10.133.192.2**</remoteIp2>

<remotePort>**2905**</remotePort>

</sctpAssociationRemoteSide>

<sctpAssociationRemoteSide name=**"RS-CDF1"**>

<remoteSystemType>**SG**</remoteSystemType>

<associationType>**diameter**</associationType>

<remoteIp1>**10.133.128.90**</remoteIp1>

<remoteIp2>**10.133.192.90**</remoteIp2>

<remotePort>**3868**</remotePort>

</sctpAssociationRemoteSide>

An SCTP association remote side has a name, remote IP addresses (IP2 is optional) and a remote port. The element remoteSystemType is always set to “SG”, the value of the element associationType are can be set to “m3ua” or “diameter”.

An SCTP association remote side can then be referenced by name in an SCTP association remote side profile or in an M3UA route. SCTP association remote side profiles are used when the Signaling Layer server does not M3UA routes – that’s the case for Diameter servers.

Defining SCTP association remote side profiles:

<sctpAssociationRemoteSideProfiles>

<sctpAssociationRemoteSideProfile name=**"CDF\_SCTP"**>

<sctpAssociationRemoteSideWrapper>

<sctpAssociationRemoteSide>**RS-CDF1**</sctpAssociationRemoteSide>

</sctpAssociationRemoteSideWrapper>

<sctpAssociationRemoteSideWrapper>

<sctpAssociationRemoteSide>**RS-CDF2**</sctpAssociationRemoteSide>

</sctpAssociationRemoteSideWrapper>

</sctpAssociationRemoteSideProfile>

</sctpAssociationRemoteSideProfiles>

Defining M3UA profiles and routes:

<m3uaProfiles>

<m3uaProfile name=**"M3UA\_ALL"**>

<routingContext>**5**</routingContext>

<ss7SignalingMode>**itu14**</ss7SignalingMode>

<m3uaRoutes>

<m3uaRoute name=**"MSS0"**>

<pointCode>**280**</pointCode>

<primaryAssociation>**RS-MSS0A**</primaryAssociation>

<secondaryAssociation>**RS-MSS0B**</secondaryAssociation>

</m3uaRoute>

<m3uaRoute name=**"MSS1"**>

<pointCode>**281**</pointCode>

<primaryAssociation>**RS-MSS1A**</primaryAssociation>

<secondaryAssociation>**RS-MSS1B**</secondaryAssociation>

</m3uaRoute>

An M3UA profile consists of multiple M3UA routes. A route defines the primary and secondary SCTP associations towards a point code, so the value between the primaryAssociation and secondaryAssociation tags must match a name attribute of an sctpAssociationRemoteSide tag seen above.

SCTP configuration of a Signaling Layer server is manifested in its SCTP association local side setting. This setting sets the local SIGTRAN addresses, and references either an M3UA profile (for CAMEL and/or MAP serving servers, so sful, sless and hlr SL servers will all have M3UA profile set here) or an SCTP association remote side profile (the sless domain will handle Diameter traffic, so sless domain will have SCTP association remote side configured). Note that an SL server can have multiple SCTP association local sides.

Example of an SL server with M3UA profile:

<signalingLayerServers>

<signalingLayerServer name=**"sfulSL01a"**>

( … )

<sctpAssociationLocalSides>

<sctpAssociationLocalSide>

<sigtranIp1>**10.133.1.1**</sigtranIp1>

<sigtranIp2>**10.133.1.2**</sigtranIp2>

<port>**2905**</port>

<side>**client**</side>

<m3uaProfile>**M3UA\_ALL**</m3uaProfile>

</sctpAssociationLocalSide>

</sctpAssociationLocalSides>

( … )

</signalingLayerServer>

##### SCCP

SCCP configuration is split into local and remote profiles. The local profile defines the local subsystems and local global title addresses. Note that an IM-SCF domain can have multiple subsystems and GT addresses but in current installation only one subsystem and GT is used per IM-SCF domain.

<sccpLocalProfile>

<localNetworkIndicator>**national**</localNetworkIndicator>

<removePcWhenRouteOnGt>**true**</removePcWhenRouteOnGt>

<localSubSystems>

<localSubSystem alias=**"IMSCF-SSN"**>

<subSystemNumber>**146**</subSystemNumber>

</localSubSystem>

</localSubSystems>

<localGtAddresses>

<localGtAddress alias=**"IMSCF-GT"**>

<globalTitle>**36309879049**</globalTitle>

<subSystemNumber>**146**</subSystemNumber>

<gtIndicator>**4**</gtIndicator>

<gtNoa>**4**</gtNoa>

<gtNumberingPlan>**1**</gtNumberingPlan>

<gtTranslationType>**0**</gtTranslationType>

</localGtAddress>

</localGtAddresses>

</sccpLocalProfile>

The SCCP remote profile defines the remote systems and their addressing. Remote systems which are addressed by point code and subsystem number are listed under remoteSubSystemPointCodeAddresses:

<sccpRemoteProfile>

<remoteSubSystemPointCodeAddresses>

<remoteSubSystemPointCode alias=**"HLRFE0"**>

<subSystemNumber>**6**</subSystemNumber>

<pointCode>**216**</pointCode>

<networkIndicator>**national**</networkIndicator>

</remoteSubSystemPointCode>

( … )

</remoteSubSystemPointCodeAddresses>

Remote systems addressed by global title are listed under remoteGtAddresses:

<sccpRemoteProfile>

<remoteSubSystemPointCodeAddresses>

( … )

</remoteSubSystemPointCodeAddresses>

<remoteGtAddresses>

<remoteGtAddress alias=**"MSS0"**>

<globalTitle>**36309489980**</globalTitle>

<subSystemNumber>**146**</subSystemNumber>

<gtIndicator>**4**</gtIndicator>

<gtNoa>**4**</gtNoa>

<gtNumberingPlan>**1**</gtNumberingPlan>

<gtTranslationType>**0**</gtTranslationType>

<networkIndicator>**national**</networkIndicator>

<pointCode>**280**</pointCode>

</remoteGtAddress>

( … )

</remoteGtAddresses>

You can configure Global Title routing right after remote GT addresses:

<sccpRemoteProfile>

<remoteSubSystemPointCodeAddresses>

( … )

</remoteSubSystemPointCodeAddresses>

<remoteGtAddresses>

( … )

</remoteGtAddresses>

<gtRouting>

<primaryGttPointCode>**281**</primaryGttPointCode>

<secondaryGttPointCode>**282**</secondaryGttPointCode>

<distribution>**loadbalance**</distribution>

</gtRouting>

</sccpRemoteProfile>

The elements below gtRouting configure GT translation. The primaryGttPointCode element is mandatory, its value is the point code which can provide global title translation. You can specify another GTT-capable point code in secondaryGttPointCode. The value of the element distribution can be either “loadbalance” or “failover” specifying how to query the point codes set above.

#### CAMEL configuration

Configuring CAMEL parameters of an IM-SCF domain is basically means defining CAP modules. CAP modules process and construct CAMEL messages which are received from and sent to the telco network.

CAP modules reference media resources, which are listed separately:

<mediaResources>

<mediaResource name=**"MRF"**>

<alias>**mrf**</alias>

<addressDigits>**03**</addressDigits>

<natureOfAddress>**national**</natureOfAddress>

<numberingPlan>**ISDN**</numberingPlan>

<bothwayThroughConnectionInd>**bothwayPathRequired**</bothwayThroughConnectionInd>

</mediaResource>

<mediaResource name=**"MRF\_FREE"**>

<alias>**mrf1**</alias>

<addressDigits>**03**</addressDigits>

<natureOfAddress>**national**</natureOfAddress>

<numberingPlan>**ISDN**</numberingPlan>

<bothwayThroughConnectionInd>**bothwayPathNotRequired**</bothwayThroughConnectionInd>

</mediaResource>

</mediaResources>

CAP modules are then defined below the “capModules” tag, one “capModule” tag for each module.

<capModules>

<capModule name=**"cap2Module"**>

<localGt>**IMSCF-GT**</localGt>

<inTriggering>

( … )

</inTriggering>

<generalProperties>

( … )

</generalProperties>

<sipProperties />

<mediaResources>

( … )

</mediaResources>

<inviteErrorHandlers>

( … )

</inviteErrorHandlers>

</capModule>

( … )

</capModules>

As you can see, a CAP module has a name and a local address. In the example above the local address is given in the “localGT”, its value must reference a localGtAddress in the SCCP local profile. Of course the local address can be given by referencing a localSubSystem in the local SCTP profile:

<capModules>

<capModule name=**"cap2Module"**>

<localSsn>**IMSCF-SSN**</localSsn>

<inTriggering>

( … )

The “inTriggering” part specifies what triggering should be used in stateful services. It specifies the events to subscribe to in the outgoing requestReportBCSMEvent message and the monitoring type as well (interrupted or notify-and-continue).

The settings under the “generalProperties” tag configure various properties of call handling and timeouts:

* reset timer intervals in seconds
* activityTest interval in seconds (after how many seconds of inactivity of a call must an activityTest message to be sent)
* asReactionTimeoutSec: how many seconds after the INVITE should the SIP AS answer
* maxCallLengthMinutes: the longest time in minutes a call can last

At “mediaResources” the media resources shown above are referenced.

The “inviteErrorHandlers” part describes what should IM-SCF do in case when the AS replies to SIP INVITE with an error. The behavior can be separated by service key as well.

#### MAP configuration

MAP configuration is about to define MAP modules in IM-SCF:

<mapModules>

<mapModule name=**"mapModule"**>

<localSsn>**IMSCF-SSN**</localSsn>

<mapGsmScfAddress>

<numberingPlan>**ISDN**</numberingPlan>

<noa>**international**</noa>

<address>**36309879049**</address>

</mapGsmScfAddress>

<mapTimeoutSec>**8**</mapTimeoutSec>

</mapModule>

</mapModules>

A MAP module has a unique name and a local address. In the example above the local address is defined by referencing a localSubSystem in the SCTP local profile but it could reference a localGtAddress as well:

<mapModules>

<mapModule name=**"mapModule"**>

<localGt>**IMSCF-GT**</localGt>

( … )

The mapGsmScfAddress specifies the GSM-SCF address to write to the outgoing anyTimeInterrogation MAP message. The “mapTimeoutSec” specifies the timeout in seconds to wait for the answer the outgoing MAP message.

#### DIAMETER Gateway configuration

Multiple DIAMETER Gateway modules can be defined to handle DIAMETER traffic towards the roamingsms service. An example DIAMETER Gateway module definition is shown below.

<diameterGatewayModules>

<diameterGatewayModule name=**"roamingSmsDiameterGwModule"**>

<acceptUndefinedPeer>**true**</acceptUndefinedPeer>

<duplicationProtection>**true**</duplicationProtection>

<duplicateTimer>**240000**</duplicateTimer>

<duplicateSize>**5000**</duplicateSize>

<queueSize>**10000**</queueSize>

<messageTimeoutMs>**2000**</messageTimeoutMs>

<stopTimeoutMs>**10000**</stopTimeoutMs>

<ceaTimeoutMs>**2000**</ceaTimeoutMs>

<iacTimeoutMs>**120000**</iacTimeoutMs>

<dwaTimeoutMs>**2000**</dwaTimeoutMs>

<dpaTimeoutMs>**5000**</dpaTimeoutMs>

<recTimeoutMs>**10000**</recTimeoutMs>

<sessionTimeoutSec>**60**</sessionTimeoutSec>

<threadPools>

<threadGroupSize>**64**</threadGroupSize>

<processingMessageTimerSize>**1**</processingMessageTimerSize>

<dplicationMessageTimerSize>**1**</dplicationMessageTimerSize>

<redirectMessageTimerSize>**1**</redirectMessageTimerSize>

<peerOverloadTimerSize>**1**</peerOverloadTimerSize>

<connectionTimerSize>**1**</connectionTimerSize>

<statisticTimerSize>**1**</statisticTimerSize>

<applicationSessionSize>**10**</applicationSessionSize>

</threadPools>

<productName>**diametergw**</productName>

<originRealm>**tsdp.telekom.hu**</originRealm>

<originApplicationId>

<vendorId>**42215**</vendorId>

<authApplId>**4**</authApplId>

<acctApplId>**4**</acctApplId>

</originApplicationId>

<destinationRealm>**gycdf.telekom.hu**</destinationRealm>

<destinationApplicationId>

<vendorId>**0**</vendorId>

<authApplId>**4**</authApplId>

<acctApplId>**0**</acctApplId>

</destinationApplicationId>

<sctpAssociationRemoteSideProfile>**CDF\_SCTP**</sctpAssociationRemoteSideProfile>

</diameterGatewayModule>

</diameterGatewayModules>

The most important property is sctpAssociationRemoteSideProfile which refers to an SCTP association remote side profile shown in section SCTP and M3UA. The other properties configure timeouts, thread pool sizes and DIAMETER-related properties.

Not strictly DIAMETER-related, but involved in DIAMETER Gateway functionality the list of HTTP application servers to where DIAMETER queries are routed to. HTTP AS can be defined at “httpApplicationServers” tag:

<httpApplicationServers>

<httpApplicationServerGroup name=**"tsdp5b\_roamingsms\_all"**>

<connectTimeoutMs>**500**</connectTimeoutMs>

<readTimeoutMs>**5500**</readTimeoutMs>

<reenableTimeMs>**15000**</reenableTimeMs>

<url>**http://192.168.1.112:8811/roamingsms\_all\_20140731/**</url>

<url>**http://192.168.1.121:8212/roamingsms\_all\_20140731/**</url>

</httpApplicationServerGroup>

( … )

</httpApplicationServers>

#### SIP configuration

SIP configuration in IM-SCF means configuring the SIP AS endpoints. SIP AS endpoints are grouped into AS groups. A call is routed to an AS group and an AS group can have multiple SIP AS inside. The distribution of the group (load-balance, fail-over) is defined in the AS group.

<sipApplicationServers>

<heartbeatConfiguration>

<timeoutSec>**5**</timeoutSec>

<activeIntervalSec>**10**</activeIntervalSec>

<inactiveIntervalSec>**30**</inactiveIntervalSec>

</heartbeatConfiguration>

<sipApplicationServerGroups>

<sipApplicationServerGroup name=**"tsdp5f\_calltester\_camel"**>

<distribution>**loadbalance**</distribution>

<sipApplicationServer name=**"calltester\_camel-1"**>

<host>**192.168.1.111**</host>

<port>**8625**</port>

<heartbeatEnabled>**true**</heartbeatEnabled>

</sipApplicationServer>

<sipApplicationServer name=**"calltester\_camel-2"**>

<host>**192.168.1.112**</host>

<port>**8625**</port>

<heartbeatEnabled>**true**</heartbeatEnabled>

</sipApplicationServer>

( … )

</sipApplicationServerGroup>

( … )

</sipApplicationServerGroups>

</sipApplicationServers>

All SIP AS endpoints are monitored by IM-SCF if they are alive or not. This monitoring is done by sending SIP OPTIONS messages to the AS and waiting for a SIP 200 OK answer. This behavior is configured in the “heartbeatConfiguration” part:

* timeoutSec is the number of seconds to wait for the 200 OK answer. After the time is elapsed and no answer arrived, the corresponding SIP AS will be marked unavailable and IM-SCF will not route calls there until it becomes available again
* activeIntervalSec is the number of seconds elapsed between two SIP OPTIONS requests sent by IM-SCF when the SIP AS is considered to be available
* inactiveIntervalSec is the number of seconds elapsed between two SIP OPTIONS requests sent by IM-SCF when the SIP AS is unavailable

SIP AS groups and their SIP AS instances are defined under the “sipApplicationServerGroups” tag. Note that IM-SCF operational scripts help you modify these structures so this part is not advised to be edited by hand.

#### Routing configuration

The following routing can be defined for an IM-SCF domain

* Route incoming calls (InitialDPs) by service key to SIP AS group
* Route incoming TCAP BEGIN messages by application context to an appropriate module
* Route incoming DIAMETER queries to HTTP AS

##### Route calls to SIP AS group

Routing is configured by adding sipAsRoutingEntry elements under the sipAsRouting tag:

<sipAsRouting>

<sipAsRoutingEntry>

<routingCriterias>

<routingCriteria>

<serviceKeyRangeList>**1-35565**</serviceKeyRangeList>

</routingCriteria>

</routingCriterias>

<sipApplicationServerGroups>

<sipApplicationServerGroupWrapper>

<sipApplicationServerGroup>**tsdp5f\_calltester\_camel**</sipApplicationServerGroup>

</sipApplicationServerGroupWrapper>

</sipApplicationServerGroups>

</sipAsRoutingEntry>

</sipAsRouting>

A sipAsRoutingEntry has two nested elements: the routingCriterias part configures when the current rule is active (in the example above it will be active if the service key is between 1 and 35565, incusive). The second element is sipApplicationServerGroups where we list the SIP AS groups to route the call to. Note that specifying more than one sipApplicationServerGroup means the service is concatenated. Important to remark that concatenation is not supported in phase one of IM-SCF introduction but the configuration structure is prepared for it.

Managing criteria and target application server groups is easier and safer to do through the scripts shipped with IM-SCF, see Using IM-SCF configuration scripts.

##### Route TCAP BEGIN to module

<moduleRouting>

<moduleRoutingEntry>

<!-- Route CAP2 to a specific module -->

<routingCriterias>

<routingCriteria>

<applicationContext>**cap2**</applicationContext>

</routingCriteria>

</routingCriterias>

<mapModule>**cap2Module**</mapModule>

</moduleRoutingEntry>

<moduleRoutingEntry>

<routingCriterias>

<routingCriteria>

<applicationContext>**map**</applicationContext>

</routingCriteria>

</routingCriterias>

<capModule>**mapModule**</capModule>

</moduleRoutingEntry>

<moduleRoutingEntry>

<routingCriterias />

<capModule>**defaultCapModule**</capModule>

</moduleRoutingEntry>

</moduleRouting>

The example above:

* routes TCAP BEGIN messages with cap2 application context to cap2Module
* routes TCAP BEGIN messages with map application context to mapModule
* and any other TCAP BEGIN messages with defaultCapModule – which is the CAP module defined for CAMEL phase 3-4 calls

The application context can be: cap2, cap3, cap4 and map.

The criteria can be a service key range as well: this way it is possible to route a call to different modules based on service key.

##### Route DIAMETER to HTTP AS

A DIAMETER routing entry is simply a three-way relationship between a service context ID, an HTTP AS group and a DIAMETER Gateway module (see DIAMETER Gateway configuration).

<diameterRouting>

<diameterRoutingEntry>

<serviceContextIds>**ROAMINGSMS**</serviceContextIds>

<httpApplicationServerGroup>**tsdp5b\_roamingsms\_all**</httpApplicationServerGroup>

<diameterGatewayModule>**roamingSmsDiameterGwModule**</diameterGatewayModule>

</diameterRoutingEntry>

</diameterRouting>

#### Notification configuration

MBean notifications can be configured at the element “notificationConfiguration”. Notifications can be defined for various counter changes: MAP operation count, CAP operation count, DIAMETER query count. The three types of notifications each is collected by some kind of service identifier. In case of CAP operation count, it is service key (serviceKeyCounterThresholdNotifications), in case of MAP (mapCounterThresholdNotifications) it is the remote system alias and in case of DIAMETER (diameterCounterThresholdNotfications) it is the service context ID.

<notificationConfiguration>

<serviceKeyCounterThresholdNotifications>

<serviceKeyCounterThresholdNotification>

<thresholdLow>**1**</thresholdLow>

<thresholdHigh>**3**</thresholdHigh>

<notificationWhenLowFromBelow>**false**</notificationWhenLowFromBelow>

<notificationWhenLowFromAbove>**true**</notificationWhenLowFromAbove>

<notificationWhenHighFromBelow>**true**</notificationWhenHighFromBelow>

<notificationWhenHighFromAbove>**false**</notificationWhenHighFromAbove>

<notificationText>**On server ${serverName} counter ${counterName} of service key ${serviceKey} crossed the ${highLow} threshold (${thresholdValue}) from ${aboveBelow}.**</notificationText>

<counterName>**initialDpCount**</counterName>

</serviceKeyCounterThresholdNotification>

</serviceKeyCounterThresholdNotifications>

<mapCounterThresholdNotifications/>

<diameterCounterThresholdNotifications/>

</notificationConfiguration>

For example the notification configuration above will send a notification every time the number of calls drops below 1 or inceases above 3. (The two thresholds are 1 and 3. The value of element notificationWhenLowFromAbove is true, so the notification will be sent when the low threshold (1) is crossed from above. The value of element notificationWhenHighFromBelow is true, so the notification will be sent when the high threshold (3) is crossed from below.)

The notification text can contain variables:

* serverName – the name of the server instance, e.g. sfulEL02b
* counterName – the name of the counter which crossed a threshold
* serviceKey – the service the counter of which has crossed a threshold (only available for service key counter notifications)
* highLow – contains the words either “high” or “low” depending on which threshold has been crossed
* aboveBelow – contains the words either “above” or “below” depending on the threshold has been crossed from above or from below
* thresholdValue – the configured threshold which has just been crossed
* serviceIdentifier – the alias of the remote system (service), e.g. “HLRFE1” or “ROAMINGSMS” (only available for MAP or DIAMETER counter notifications)

The three types of notifications obviously can be configured to watch different counter values. In case of CAP, the following counter names are available:

* activityTestRequestCount
* activityTestResponseCount
* applyChargingCount
* applyChargingReportCount
* callInformationReportRequestCount
* callInformationReportResponseCount
* cancelCount
* connectCount
* connectToResourceCount
* continueCount
* continueWithArgumentCount
* disconnectForwardConnectionCount
* disconnectForwardConnectionWithArgumentCount
* disconnectLegCount
* eventReportBcsmCount
* furnishChargingInformationCount
* initialDpCount
* initiateCallAttemptRequestCount
* moveLegRequestCount
* moveLegResponseCount
* playAnnouncementCount
* promptAndCollectUserInformationCount
* promptAndCollectUserInformationResultCount
* releaseCallCount
* requestReportBcsmEventCount
* resetTimerCount
* returnResultLastCount
* specializedResourceReportCount
* splitLegCount
* tcapReceivedCount
* tcapBeginReceivedCount
* tcapContinueReceivedCount
* tcapEndReceivedCount
* tcapAbortReceivedCount
* tcapSentCount
* tcapBeginSentCount
* tcapContinueSentCount
* tcapEndSentCount
* tcapAbortSentCount

The following MAP counters can be watched:

* anyTimeInterrogationCount
* anyTimeInterrogationResultCount
* tcapReceivedCount
* tcapBeginReceivedCount
* tcapContinueReceivedCount
* tcapEndReceivedCount
* tcapAbortReceivedCount
* tcapSentCount
* tcapBeginSentCount
* tcapContinueSentCount
* tcapEndSentCount
* tcapAbortSentCount

In case of DIAMETER the following counters are available:

* balanceQueryReceivedCount
* balanceQueryAnsweredCount
* debitQueryReceivedCount
* debitQueryAnsweredCount

#### Instance configuration

IM-SCF instances can be defined under the “servers” tag

<servers>

<signalingLayerServers>

<signalingLayerServer name=**"sfulSL01a"**>

<poolConfig>**SLPoolConfig**</poolConfig>

<connectivity>

<internalCommunicationAddress>

<host>**192.168.2.21**</host>

<port>**12113**</port>

</internalCommunicationAddress>

<administrationAddress>

<host>**192.168.2.21**</host>

<port>**12111**</port>

</administrationAddress>

</connectivity>

<sctpAssociationLocalSides>

<sctpAssociationLocalSide>

( … see SIGTRAN configuration … )

</sctpAssociationLocalSide>

</sctpAssociationLocalSides>

<pointCode>**395**</pointCode>

<mtpDeliveryTransferMessageThreadCount>**8**</mtpDeliveryTransferMessageThreadCount>

<sctpWorkerThreadCount>**1**</sctpWorkerThreadCount>

</signalingLayerServer>

( … )

</signalingLayerServers>

<executionLayerServers>

<executionLayerServer name=**"sfulEL01a"**>

<poolConfig>**ELPoolConfig**</poolConfig>

<connectivity>

<internalCommunicationAddress>

<host>**192.168.2.21**</host>

<port>**32113**</port>

</internalCommunicationAddress>

<administrationAddress>

<host>**192.168.2.21**</host>

<port>**32111**</port>

</administrationAddress>

<sipListenAddress>

<host>**192.168.1.21**</host>

<port>**32112**</port>

</sipListenAddress>

</connectivity>

<tcapTransactionIdRange>

<minInclusive>**1000000**</minInclusive>

<maxInclusive>**1999999**</maxInclusive>

</tcapTransactionIdRange>

</executionLayerServer>

( … )

</executionLayerServers>

</servers>

The following properties must be set for Signaling Layer servers:

* Name
* Pool Configuration: this is a technical parameter, controls the number of threads, need not to be changed
* Internal Communication Address: SL and EL servers will communicate through this port. See Domains, machines and processes
* Administration Address: administration functions are reachable through this port
* SCTP Association Local Sides: see SIGTRAN configuration
* Point Code: the point code assigned to this SL server

The Execution Layer servers have the following additional properties (and they do not have the SCTP Association Local Sides and Point Code parameters):

* SIP Listen Address: address used for SIP communication with SIP AS
* TCAP Transaction ID range: The TCAP TID range used by this server. Note that the ranges must not overlap among the EL servers of the same IM-SCF domain

#### Overload protection

Under the “overloadProtection” tag the SL and EL servers can be configured when to enter overloaded state. When a server is in overloaded state, it intentionally does not answer incoming requests (or sends back error immediately) so it does not get to a state it could not recover from. The configuration looks like the following:

<overloadProtection>

<enabled>**true**</enabled>

<cpuOverloadThresholdPercent>**80**</cpuOverloadThresholdPercent>

<cpuMeasurementWindow>**10**</cpuMeasurementWindow>

<dataCollectionPeriodSec>**3**</dataCollectionPeriodSec>

<heapOverloadThresholdPercent>**90**</heapOverloadThresholdPercent>

<nonHeapOverloadThresholdPercent>**90**</nonHeapOverloadThresholdPercent>

</overloadProtection>

The following settings indicate that a server will be in overloaded state when the average CPU usage is above 80 percent and the 90 of the heap is already used. It should average the CPU from 10 measurements which are taken every 3 seconds. These settings are proved to be safe from earlier experiences.

#### Technical parameters

At the “lwCommParameters” section the IM-SCF configuration XML contains some technical parameters which are not likely to change in time.

### Using IM-SCF configuration scripts

Configuring IM-SCF via editing the configuration XML file is prone to errors, many mistakes can be made. Therefore IM-SCF provides some scripts to ease the most common processes in IM-SCF maintenance.

#### Basics

The functionality is assembled into one script called imscfAdmin.sh. You can get help by invoking the script with -h or --help options:

$ imscfAdmin.sh -h

or

$ imscfAdmin.sh --help

The script is basically operated by specifying the target domain and a command to execute. The commands of imscfAdmin.sh can be divided into two categories: commands which merely give info of the configuration (non-modifying) and commands which actually alter the configuration (modifying).

To issue a modifying command, an editing session must be established. An editing session belongs to a single domain, is owned by the user logged in and is created for altering the configuration on a so-called working copy. Only one editing session can be active for a specific domain. The editing session can be discarded or committed by its owner user.

When creating an editing session, the currently active configuration of the domain is copied in a so-called working copy. Further modifying and non-modifying commands are executed on this working copy. If the editing session is discarded, the working copy is deleted and the active configuration is not touched. If the editing session is committed the following happens:

1. A new configuration archive entry is created and the active and new configuration is copied there. The timestamp and the user name is saved here
2. The active configuration is overwritten by the working copy
3. The working copy is discarded

Since the imscfAdmin.sh script does not distribute the configuration files, neither refreshes IM-SCF instances, you have to do this manually by invoking the distributeConfig.sh as seen in section Editing XML files manually.

You can start an editing session by issuing

$ imscfAdmin.sh <domain> --session start

The script checks if an editing session is active and creates a new one if there was none. Gives an error if there is already an active editing session for the given domain. Creating a new editing session basically consists of placing a lock file and creating a working copy in the directory /usr/imscf/configuration/<domain>/work.

A lock file is a regular file in the directory /usr/imscf/configuration/<domain>/work. Creating a working copy essentially means copying the file /usr/imscf/configuration/<domain>/<domain>\_imscf\_config.xml to /usr/imscf/configuration/<domain>/work/<domain>\_imscf\_config\_workingcopy.xml.

The script gives an error if there is already an active editing session for domain <domain>. To discard the editing session for domain <domain> enter:

$ imscfAdmin.sh <domain> --session discard

The discard operation will delete the lock and workingcopy files created by the start operation.

Editing session can be committed by issuing:

$ imscfAdmin.sh <domain> --session commit [commit\_message]

When committing an editing session, above from applying changes, a new configuration history entry is created. Configuration history of a domain can be found in the directory /usr/imscf/configuration/domain/history. This directory contains timestamped subdirectories which also contain the user name who made the commit. The timestamped subdirectory contains two files: <domain>\_imscf\_config\_before.xml and <domain>\_imscf\_config\_after.xml. The prior contains the domain’s IM-SCF configuration when the editing session was started, the latter contains the new configuration with modifications. If the optional parameter commit\_message is present, a file named “comment” is created in the history directory with the contents of the parameter value. It is advised to put quotation marks around the message, e.g.

$ imscfAdmin.sh sless --session commit “Routed odomino to quarantine”

If you want to check if an editing session is active or not for a specific domain, enter

$ imscfAdmin.sh <domain> --session status

Note that issuing a non-modifying command does not need an active editing session. When there is no editing session started, the non-modifying commands use the active configuration of the domain for querying. When there is an active edit session, the non-modifying commands read the information from the working copy so you can check the modified version before committing.

#### Managing SIP Application Servers

As seen in SIP configuration SIP application servers are arranged into SIP application server groups. You can list the AS groups configured with the command --asGroups list:

$ imscfAdmin.sh <domain> --asGroups list [<name>]

If the optional <name> parameter is given, then only the groups which contain the parameter’s value in their names will be listed.

$ imscfAdmin.sh sful --asGroups list domino

Setting IM-SCF Environment

This is a PILOT node on pilot-oam01

The given configuration file is loaded and valid!

SIP Application Server Groups

+-------------------------------------------------+-------+-------------------------+---------------+------+-------------------+

| imscf5c\_odomino\_camel instances with LOADBALANCE | INDEX | Application Server Name | Host/IP | Port | Heartbeat Enabled |

+-------------------------------------------------+-------+-------------------------+---------------+------+-------------------+

| | 0 | odomino\_camel-1 | 192.168.1.121 | 8314 | true |

+-------------------------------------------------+-------+-------------------------+---------------+------+-------------------+

+-------------------------------------------------+-------+-------------------------+---------------+------+-------------------+

| imscf5b\_tdomino\_camel instances with LOADBALANCE | INDEX | Application Server Name | Host/IP | Port | Heartbeat Enabled |

+-------------------------------------------------+-------+-------------------------+---------------+------+-------------------+

| | 0 | tdomino\_camel-1 | 192.168.1.121 | 8214 | true |

+-------------------------------------------------+-------+-------------------------+---------------+------+-------------------+

Add a new group:

$ imscfAdmin.sh <domain> --asGroups add <group\_name> <distribution>

Where the parameters

* group\_name – the unique name of the new group
* distribution – how the calls will be distributed among the contained SIP application servers: “failover” or “loadbalance”

If there is already a group with the given groupname parameter, the script signals an error and no change is made.

To delete an existing group:

$ imscfAdmin.sh <domain> --asGroups remove <group\_name>

If there is no AS group with the name of the parameter, the script returns with error and no change is made.

To change the distribution of an existing AS group:

$ imscfAdmin.sh <domain> --asGroups edit <group\_name> <distribution>

SIP application server groups contain an ordered list of application servers. You can use the list command of the --asGroups switch to list groups with their application servers as seen above. You can manage SIP application servers inside an application server group with the --asHandling switch.

To add a SIP application server to a group use the “--asHandling add” command:

$ imscfAdmin.sh <domain> --asHandling add <group\_name> <as\_name> <host\_port> <heartbeat> [<index>]

Where the parameters are the following:

* group\_name – The name of the SIP application server group to add the new server to. Must be the name of an existing group
* as\_name – The name of the new SIP application server. Must be unique in the group the server will be added to
* host\_port – The SIP endpoint address of the new application server in HOSTNAME:PORT format. E.g. 192.168.1.131:8611, or site2-imscfc1-sip:8611
* heartbeat – If IM-SCF should check the server’s availability with SIP OPTIONS heartbeat messages. The value of the parameter can be either “yes” or “no”.
* index – This optional parameter specifies where to insert the new application server in the list. 0 means the beginning of the list, the value can be any nonnegative integer. If the value is greater than or equal to the current size of the list the new server is inserted as the last element. This append behavior is the default when the parameter is missing

To modify the SIP address of a SIP application server use the editSip subcommand:

$ imscfAdmin.sh <domain> --asHandling editSip <group\_name> <as\_name> <host\_port>

The parameters “group\_name” and “as\_name” must refer to an existing SIP AS group and a SIP AS inside it.

To switch the SIP OPTIONS heartbeat towards a SIP application server on and off, use the editHb subcommand:

$ imscfAdmin.sh <domain> --asHandling editHb <group\_name> <as\_name> <heartbeat>

The parameters “group\_name” and “as\_name” must refer to an existing SIP AS group and a SIP AS inside it. The value of the parameter “heartbeat” can be “yes” or “no” as seen above.

You can alter the order of the SIP application servers inside a group with the “move” subcommand:

$ imscfAdmin.sh <domain> --asHandling move <group\_name> <as\_name> <new\_index>

The value of the parameter <new\_index> is interpreted exactly like the parameter <index> in the --asHandling add command.

To remove a SIP application server from a group, invoke the imscfAdmin.sh script with “--asHandling remove” command

$ imscfAdmin.sh <domain> --asHandling remove <group\_name> <as\_name>

#### Configuring SIP routing

The imscfAdmin.sh script’s --asRouting switch can be used to check and modify the SIP routing by service key. You can check the routing with the listAll, listBySk, listByGroup subcommands

$ imscfAdmin.sh <domain> --asRouting listAll

$ imscfAdmin.sh <domain> --asRouting listBySk <service\_key>

$ imscfAdmin.sh <domain> --asRouting listByGroup <group\_name>

The listAll subcommand lists all SIP routing settings:

$ imscfAdmin.sh sless --asRouting listAll

Setting IM-SCF Environment

This is a PILOT node on pilot-oam01

The given configuration file is loaded and valid!

SIP Application Server Routings

+--------------------------------------------+---------------------------------------------+

| Routing Entry --- INDEX: 0 | |

+--------------------------------------------+---------------------------------------------+

| Routing Criteria Type | Routing Criteria Value |

| ServicKey Range List | 20012,28074 |

| ------------------------------------------ | ------------------------------------------- |

| Server Groups | |

| imscf5c\_odomino\_camel | |

+--------------------------------------------+---------------------------------------------+

+--------------------------------------------+---------------------------------------------+

| Routing Entry --- INDEX: 1 | |

+--------------------------------------------+---------------------------------------------+

| Routing Criteria Type | Routing Criteria Value |

| ServicKey Range List | 1-35565 |

| ------------------------------------------ | ------------------------------------------- |

| Server Groups | |

| imscf5a\_calltester\_all | |

+--------------------------------------------+---------------------------------------------+

The listBySk subcommand lists all routing entries which are configured for the given service key. Not that only the first entry will be activated, since the rule listed earlier has priority over the other rules.

$ imscfAdmin.sh sless --asRouting listBySk 28040

Setting IM-SCF Environment

This is a PILOT node on pilot-oam01

The given configuration file is loaded and valid!

SIP Application Server Routings

+--------------------------------------------+---------------------------------------------+

| Routing Entry --- INDEX: 1 | |

+--------------------------------------------+---------------------------------------------+

| Routing Criteria Type | Routing Criteria Value |

| ServicKey Range List | 1-35565 |

| ------------------------------------------ | ------------------------------------------- |

| Server Groups | |

| imscf5a\_calltester\_all | |

+--------------------------------------------+---------------------------------------------+

The listByGroup lists all rules the target of which contains the given SIP AS group.

$ imscfAdmin.sh sless --asRouting listByGroup imscf5c\_odomino\_camel

Setting IM-SCF Environment

This is a PILOT node on pilot-oam01

The given configuration file is loaded and valid!

SIP Application Server Routings

+-------------------------------------------+--------------------------------------------+

| Routing Entry --- INDEX: 0 | |

+-------------------------------------------+--------------------------------------------+

| Routing Criteria Type | Routing Criteria Value |

| ServicKey Range List | 20012,28074 |

| ----------------------------------------- | ------------------------------------------ |

| Server Groups | |

| imscf5c\_odomino\_camel | |

+-------------------------------------------+--------------------------------------------+

To add a new routing rule use the “--asRouting add” command:

$ imscfAdmin.sh <domain> --asRouting add <service\_key\_range> <group\_names> [<index>]

* service\_key\_range – comma-separated list of service keys and service key ranges. Service key ranges are in the form “X-Y” where X must be smaller than Y. For example, the following value includes all service keys between 3 and 10: “3,4,5-7,8-10”
* group\_names – comma-separated list of SIP application server group names. The calls with the service keys above will be routed to these groups in the order they appear here, so specifying more than one group means service concatenation
* index – this optional parameter specifies where to insert the new rule in the list. Zero means the beginning of the list, the value can be any nonnegative integer. If the value is greater than or equal to the current size of the rule list the new server is inserted as the last element. This append behavior is the default when the parameter is missing

To remove a routing rule use the “remove” subcommand. The index specified can be determined by one of the list commands:

$ imscfAdmin.sh <domain> --asRouting remove <index>

To alter the list of the rules use the “--asRouting move” command:

$ imscfAdmin.sh <domain> --asRouting move <actual\_index> <new\_index>

The value of “actual\_index” is to be determined with the listing commands. The value of new\_index must be given according to the rules applied to the “index” parameter in the “--asRouting add” command.

There is a special “edit” command to quickly change the routing of a service key (or service key range) instead of deleting an existing rule and creating a new one in two steps:

$ imscfAdmin.sh <domain> --asRouting edit <service\_key\_range> <group\_names>

The command looks for exact <service\_key\_range> routing rule match and if there is one, then <group\_names> will be set as new target SIP application server groups.

#### Configuring DIAMETER

Configuring Diameter with the imscfAdmin.sh script means managing HTTP application servers and routing incoming DIAMETER requests by service context ID to HTTP application servers and Diameter modules.

HTTP application servers are organized into groups like the SIP application servers. Managing HTTP application server groups can be managed using the standard add-edit-remove commands under the switch --diameterHandling.

You can list the currently configured HTTP application servers with the HttpAs subcommand of the --configListing switch:

$ imscfAdmin.sh <domain> --configListing HttpAs  
HTTP Application Server Groups

Name:imscf5b\_roamingsms\_all, Index:0

+--------------------+-----------------+---------------------+

| Connect Timeout MS | Read Timeout MS | Reenable Timeout MS |

+--------------------+-----------------+---------------------+

| 500 | 5500 | 15000 |

+--------------------+-----------------+---------------------+

imscf5b\_roamingsms\_all - URLs

+----------------------------------------------------+

| URLs |

+----------------------------------------------------+

| http://192.168.1.112:8811/roamingsms\_all\_20140731/ |

| http://192.168.1.121:8212/roamingsms\_all\_20140731/ |

+----------------------------------------------------+

To add a new HTTP application server group, issue:

$ imscfAdmin.sh <domain> --diameterHandling addHttpAsGroup <group\_name> <url\_list>

Where

* group\_name – a unique name for the new group
* url\_list – a comma-separated list of the HTTP endpoints of the HTTP application servers inside the group

To modify the HTTP application server endpoints in a HTTP application server group, use the “—diameterHandling editHttpAsGroup” command. Note that HTTP application servers cannot be added to or removed from a group individually.

$ imscfAdmin.sh <domain> --diameterHandling editHttpAsGroup <group\_name> <url\_list>

To remove a HTTP application server group, issue:

$ imscfAdmin.sh <domain> --diameterHandling removeHttpAsGroup <group\_name>

To configure the routing that which DIAMETER service context ID is routed to which HTTP AS group and Diameter Module use the --diameterHandling switch with the addDiameterRouting, editDiameterRouting and removeDiameterRouting subcommands.

You can check the currently active rules with the “—configListing DiameterRoutings” command:

$ imscfAdmin.sh <domain> --configListing DiameterRoutings  
Diameter Routings

+-------+---------------------+-------------------------------+----------------------------+

| INDEX | Service Context IDs | HTTP Application Server Group | Diameter Gateway Module |

+-------+---------------------+-------------------------------+----------------------------+

| 0 | ROAMINGSMS | imscf5b\_roamingsms\_all | roamingSmsDiameterGwModule |

+-------+---------------------+-------------------------------+----------------------------+

To add a new DIAMETER routing rule:

$ imscfAdmin.sh <domain> --diameterHandling addDiameterRouting <context\_id> <group\_name> <diameter\_module>

Where the parameters are the following:

* context\_id – a service context ID which defines a service. For example “ROAMINGSMS”
* group\_name – a name of an existing HTTP application server group
* diameter\_module – a name of an existing Diameter Module. Diameter Modules can be configured as seen in DIAMETER Gateway configuration and you can list the with the --configListing switch. For example “roamingSmsDiameterGwModule”

You can list the available Diameter modules with the “—configListing DiameterModules” command:

$ imscfAdmin.sh <domain> --configListing DiameterModules  
DiameterModules

Diameter Gateway Modules

+----------------------------+--------------+ (…)

| Name | Product Name | (…)

+----------------------------+--------------+ (…)

| roamingSmsDiameterGwModule | diametergw | (…)

+----------------------------+--------------+ (…)

-----------------+--------------------------------------------------+ (…)

Origin Realm | Origin Application ID | (…)

-----------------+--------------------------------------------------+ (…)

imscf.restcomm.com | Vendor ID: 42215, Auth App ID: 4, Acct App ID: 4 | (…)

-----------------+--------------------------------------------------+ (…)

-------------------+----------------------------------------------+--------------------------------+

Destination Realm | Destination Application ID | SCTP Assoc Remote Side Profile |

-------------------+----------------------------------------------+--------------------------------+

gycdf.telekom.hu | Vendor ID: 0, Auth App ID: 4, Acct App ID: 0 | CDF\_SCTP |

-------------------+----------------------------------------------+--------------------------------+

To edit an existing routing rule, use

$ imscfAdmin.sh <domain> --diameterHandling editDiameterRouting <index> <group\_name>

This command changes the target HTTP application server group of an existing rule. The rule to modify is specified by its index (the index is zero-based and can be queried using one of the listing methods). The new HTTP application server group to route to is specified by the parameter <group\_name>.

To remove a routing entry, issue:

$ imscfAdmin.sh <domain> --diameterHandling removeDiameterRouting <index>

The command deletes the rule at the specified (zero-based) index.

#### Other commands

To list all available commands of imscfAdmin.sh, invoke the script with the parameter --listCommands:

$ imscfAdmin.sh --listCommands

You can list all domains configured:

$ imscfAdmin.sh –listDomains

To validate the configuration in the current working copy, use the --validate switch. Note that only valid configuration can be committed.

$ imscfAdmin.sh <domain> --validate

To list various parts of the configuration use the --configListing switch:

$ imscfAdmin.sh <domain> --configListing <config\_list\_param>

Where the <config\_list\_param> specifies the entities which you want to list from the configuration. The valid values are the following:

**SctpAssoc** – To list the SCTP associations configured as described in section SCTP and M3UA. Example output:

+-----------+--------------+-------------+----------------+----------------+-------------+

| Name | Rem.Sys.Type | Assoc. Type | Remote IP 1 | Remote IP 2 | Remote Port |

+-----------+--------------+-------------+----------------+----------------+-------------+

| RS-HLRFE0 | SG | m3ua | 10.133.147.17 | 10.133.211.17 | 2905 |

| RS-HLRFE1 | SG | m3ua | 10.133.151.17 | 10.133.215.17 | 2905 |

| RS-HLRFE2 | SG | m3ua | 10.133.144.33 | 10.133.208.33 | 2905 |

| RS-MSS0A | SG | m3ua | 10.133.128.2 | 10.133.192.2 | 2905 |

| RS-MSS0B | SG | m3ua | 10.133.128.6 | 10.133.192.6 | 2905 |

| RS-MSS1A | SG | m3ua | 10.133.129.2 | 10.133.193.2 | 2905 |

| RS-MSS1B | SG | m3ua | 10.133.129.6 | 10.133.193.6 | 2905 |

| RS-MSS2A | SG | m3ua | 10.133.131.10 | 10.133.195.10 | 2905 |

| RS-MSS2B | SG | m3ua | 10.133.131.14 | 10.133.195.14 | 2905 |

| RS-MSS3A | SG | m3ua | 10.133.132.2 | 10.133.196.2 | 2905 |

| RS-MSS3B | SG | m3ua | 10.133.132.6 | 10.133.196.6 | 2905 |

| RS-MSS4A | SG | m3ua | 10.133.133.2 | 10.133.197.2 | 2905 |

| RS-MSS4B | SG | m3ua | 10.133.133.6 | 10.133.197.6 | 2905 |

| RS-MSS5A | SG | m3ua | 10.133.134.2 | 10.133.198.2 | 2905 |

| RS-MSS5B | SG | m3ua | 10.133.134.6 | 10.133.198.6 | 2905 |

| RS-MSS6A | SG | m3ua | 10.133.135.2 | 10.133.199.2 | 2905 |

| RS-MSS6B | SG | m3ua | 10.133.135.6 | 10.133.199.6 | 2905 |

| RS-CDF1 | SG | diameter | 10.133.128.90 | 10.133.192.90 | 3868 |

| RS-CDF2 | SG | diameter | 10.133.128.110 | 10.133.192.110 | 3868 |

+-----------+--------------+-------------+----------------+----------------+-------------+

**M3uaProfiles** – To list the available M3UA profiles configured as described in section SCTP and M3UA. Example output:

M3uaProfile: M3UA\_ALL, Routing Context: 5, SS7 Singaling Mode: ITU\_14

+--------+-----+---------------------+-----------------------+

| Name | PC | Primary Association | Secondary Association |

+--------+-----+---------------------+-----------------------+

| HLRFE0 | 216 | RS-HLRFE0 | |

| HLRFE1 | 202 | RS-HLRFE1 | |

| HLRFE2 | 203 | RS-HLRFE2 | |

| MSS0 | 280 | RS-MSS0A | RS-MSS0B |

| MSS1 | 281 | RS-MSS1A | RS-MSS1B |

| MSS2 | 282 | RS-MSS2A | RS-MSS2B |

| MSS3 | 283 | RS-MSS3A | RS-MSS3B |

| MSS4 | 284 | RS-MSS4A | RS-MSS4B |

| MSS5 | 285 | RS-MSS5A | RS-MSS5B |

| MSS6 | 286 | RS-MSS6A | RS-MSS6B |

+--------+-----+---------------------+-----------------------+

**SccpProfiles** – To list the SCCP local and remote profiles configured as described in section SCCP. Example output:

+--------------------------------------------------------------------------------------------+

| SCCP Local Profile --- Local Network Indicator: NATIONAL, Remove PC When Route On GT: true |

+--------------------------------------------------------------------------------------------+

SCCP Local Subsystems

+-----------+------------------+

| Alias | Subsystem Number |

+-----------+------------------+

| IMSCF-SSN | 146 |

+-----------+------------------+

SCCP Local GT Addresses

+----------+--------------+------------------+--------------+--------+-------------------+---------------------+

| Alias | Global Title | Subsystem Number | GT Indicator | GT Noa | GT Numbering Plan | GT Translation Type |

+----------+--------------+------------------+--------------+--------+-------------------+---------------------+

| IMSCF-GT | 36309879052 | 146 | 4 | 4 | 1 | 0 |

+----------+--------------+------------------+--------------+--------+-------------------+---------------------+

+---------------------+

| SCCP Remote Profile |

+---------------------+

SCCP Remote Subsystem PC Addresses

+--------+------------------+-----+-------------------+

| Alias | Subsystem Number | PC | Network Indicator |

+--------+------------------+-----+-------------------+

| HLRFE0 | 6 | 216 | NATIONAL |

| HLRFE1 | 6 | 202 | NATIONAL |

| HLRFE2 | 6 | 203 | NATIONAL |

+--------+------------------+-----+-------------------+

SCCP Remote GT Addresses

+-----------+--------------+---------+---------+--------+--------------+----------------+-------------------+-----+

| Alias | Global Title | Subsys. | GT Ind. | GT Noa | GT Numb. Pl. | GT Trans. Type | Network Indicator | PC |

+-----------+--------------+---------+---------+--------+--------------+----------------+-------------------+-----+

| MSS0 | 36309489980 | 146 | 4 | 4 | 1 | 0 | NATIONAL | 280 |

| MSS1 | 36309489981 | 146 | 4 | 4 | 1 | 0 | NATIONAL | 281 |

| MSS2 | 36309489982 | 146 | 4 | 4 | 1 | 0 | NATIONAL | 282 |

| MSS3 | 36309489983 | 146 | 4 | 4 | 1 | 0 | NATIONAL | 283 |

| MSS4 | 36309489984 | 146 | 4 | 4 | 1 | 0 | NATIONAL | 284 |

| MSS5 | 36309489985 | 146 | 4 | 4 | 1 | 0 | NATIONAL | 285 |

| MSS6 | 36309489986 | 146 | 4 | 4 | 1 | 0 | NATIONAL | 286 |

| HLRFE0FNR | 36309489916 | 222 | 4 | 4 | 1 | 250 | NATIONAL | 216 |

| HLRFE1FNR | 36309489900 | 222 | 4 | 4 | 1 | 250 | NATIONAL | 202 |

| HLRFE2FNR | 36309489900 | 222 | 4 | 4 | 1 | 250 | NATIONAL | 203 |

+-----------+--------------+---------+---------+--------+--------------+----------------+-------------------+-----+

GT Routing

+------------+--------------+--------------+

| Primary PC | Secondary PC | Distribution |

+------------+--------------+--------------+

| 280 | 280 | LOADBALANCE |

+------------+--------------+--------------+

**MediaResources** – To list the media resources available for CAP modules configured as seen in section CAMEL configuration. Example output:

+----------+-------+----------------+----------+----------------+---------------------------+

| Name | Alias | Address Digits | Noa | Numbering Plan | Bothway |

+----------+-------+----------------+----------+----------------+---------------------------+

| MRF | mrf | 03 | NATIONAL | ISDN | BOTHWAY\_PATH\_REQUIRED |

| MRF\_FREE | mrf1 | 03 | NATIONAL | ISDN | BOTHWAY\_PATH\_NOT\_REQUIRED |

+----------+-------+----------------+----------+----------------+---------------------------+

**ModuleRoutings** – To list the module routings configured as described in section Route TCAP BEGIN to module. Example output:

+------------------+----------------------------+

| Module | Routing Criterias |

+------------------+----------------------------+

| cap2Module | Application context: CAP\_2 |

| mapModule | Application context: MAP |

| defaultCapModule | |

+------------------+----------------------------+

**Servers** – To list the configured SL and EL servers. Example output:

+--------------------------------+

| Servers of the Signaling Layer |

+--------------------------------+

slessSL01a

+--------------------------------+------------------------+-----+--------------------------------+

| Internal Communication Address | Administration Address | PC | Diameter Gateway Origin Host |

+--------------------------------+------------------------+-----+--------------------------------+

| 192.168.2.21:11113 | 192.168.2.21:11111 | 393 | pilot-mobile03.imscf.restcomm.com |

+--------------------------------+------------------------+-----+--------------------------------+

slessSL01a - SCTP Associations Local Side

+----------------+----------------+------+--------+--------------+--------------------------------------+

| Sigtran IP 1 | Sigtran IP 2 | Port | Side | M3ua Profile | SCTP Association Remote Side Profile |

+----------------+----------------+------+--------+--------------+--------------------------------------+

| 10.133.128.154 | 10.133.192.154 | 2905 | client | M3UA\_ALL | |

| 10.133.128.154 | 10.133.192.154 | 3868 | client | | CDF\_SCTP |

+----------------+----------------+------+--------+--------------+--------------------------------------+

slessSL02a

+--------------------------------+------------------------+-----+--------------------------------+

| Internal Communication Address | Administration Address | PC | Diameter Gateway Origin Host |

+--------------------------------+------------------------+-----+--------------------------------+

| 192.168.2.22:11213 | 192.168.2.22:11211 | 394 | pilot-mobile04.imscf.restcomm.com |

+--------------------------------+------------------------+-----+--------------------------------+

slessSL02a - SCTP Associations Local Side

+----------------+----------------+------+--------+--------------+--------------------------------------+

| Sigtran IP 1 | Sigtran IP 2 | Port | Side | M3ua Profile | SCTP Association Remote Side Profile |

+----------------+----------------+------+--------+--------------+--------------------------------------+

| 10.133.128.158 | 10.133.192.158 | 2905 | client | M3UA\_ALL | |

| 10.133.128.158 | 10.133.192.158 | 3868 | client | | CDF\_SCTP |

+----------------+----------------+------+--------+--------------+--------------------------------------+

+--------------------------------+

| Servers of the Execution Layer |

+--------------------------------+

+----------------+--------------------------------+------------------------+--------------------+----------------------+

| EL Server Name | Internal Communication Address | Administration Address | SIP Listen Address | Transaction ID Range |

+----------------+--------------------------------+------------------------+--------------------+----------------------+

| slessEL01a | 192.168.2.21:31113 | 192.168.2.21:31111 | 192.168.1.21:31112 | 1999999-1999999 |

| slessEL01b | 192.168.2.21:31123 | 192.168.2.21:31121 | 192.168.1.21:31122 | 2999999-2999999 |

| slessEL02a | 192.168.2.22:31213 | 192.168.2.22:31211 | 192.168.1.22:31212 | 3999999-3999999 |

| slessEL02b | 192.168.2.22:31223 | 192.168.2.22:31221 | 192.168.1.22:31222 | 4999999-4999999 |

+----------------+--------------------------------+------------------------+--------------------+----------------------+

**DiameterModules** – To list the Diameter Modules configured as described in section DIAMETER Gateway configuration. Example output:

Diameter Gateway Modules

+----------------------------+--------------+-----------------+--------------------------------------------------+ …

| Name | Product Name | Origin Realm | Origin Application ID | …

+----------------------------+--------------+-----------------+--------------------------------------------------+ …

| roamingSmsDiameterGwModule | diametergw | imscf.restcomm.com | Vendor ID: 42215, Auth App ID: 4, Acct App ID: 4 | …

+----------------------------+--------------+-----------------+--------------------------------------------------+ …

(right part of the table)

-------------------+----------------------------------------------+--------------------------------+

Destination Realm | Destination Application ID | SCTP Assoc Remote Side Profile |

-------------------+----------------------------------------------+--------------------------------+

gycdf.telekom.hu | Vendor ID: 0, Auth App ID: 4, Acct App ID: 0 | CDF\_SCTP |

-------------------+----------------------------------------------+--------------------------------+

**DiameterRoutings** – To list the Diameter routings configured either by script or by hand. Example output:

Diameter Routings

+-------+---------------------+-------------------------------+----------------------------+

| INDEX | Service Context IDs | HTTP Application Server Group | Diameter Gateway Module |

+-------+---------------------+-------------------------------+----------------------------+

| 0 | ROAMINGSMS | imscf5b\_roamingsms\_all | roamingSmsDiameterGwModule |

+-------+---------------------+-------------------------------+----------------------------+

**HttpAs** – To list the configured HTTP application server groups and their HTTP application server endpoints. Example output:

HTTP Application Server Groups

Name:imscf5b\_roamingsms\_all, Index:0

+--------------------+-----------------+---------------------+

| Connect Timeout MS | Read Timeout MS | Reenable Timeout MS |

+--------------------+-----------------+---------------------+

| 500 | 5500 | 15000 |

+--------------------+-----------------+---------------------+

imscf5b\_roamingsms\_all - URLs

+----------------------------------------------------+

| URLs |

+----------------------------------------------------+

| http://192.168.1.112:8811/roamingsms\_all\_20140731/ |

| http://192.168.1.121:8212/roamingsms\_all\_20140731/ |

+----------------------------------------------------+

**SipAs** – To list the SIP application server groups. Example output:

SIP Application Server Groups

+----------------------------------------------------+-------+-------------------------+---------------+------+-------------------+

| imscf5a\_calltester\_all instances with LOADBALANCE | INDEX | Application Server Name | Host/IP | Port | Heartbeat Enabled |

+----------------------------------------------------+-------+-------------------------+---------------+------+-------------------+

| | 0 | calltester\_all-1 | 192.168.1.111 | 8115 | true |

| | 1 | calltester\_all-2 | 192.168.1.112 | 8115 | true |

+----------------------------------------------------+-------+-------------------------+---------------+------+-------------------+

+----------------------------------------------------+-------+-------------------------+---------------+------+-------------------+

| imscf5a\_calltester\_camel instances with LOADBALANCE | INDEX | Application Server Name | Host/IP | Port | Heartbeat Enabled |

+----------------------------------------------------+-------+-------------------------+---------------+------+-------------------+

| | 0 | calltester\_camel-1 | 192.168.1.121 | 8125 | true |

| | 1 | calltester\_camel-2 | 192.168.1.121 | 8125 | true |

+----------------------------------------------------+-------+-------------------------+---------------+------+-------------------+

+----------------------------------------------------+-------+-------------------------+---------------+------+-------------------+

| imscf5c\_odomino\_camel instances with LOADBALANCE | INDEX | Application Server Name | Host/IP | Port | Heartbeat Enabled |

+----------------------------------------------------+-------+-------------------------+---------------+------+-------------------+

| | 0 | odomino\_camel-1 | 192.168.1.121 | 8314 | true |

+----------------------------------------------------+-------+-------------------------+---------------+------+-------------------+

### Deploying new IM-SCF binaries

Like any other software, the IM-SCF system’s binaries will be refreshed from time to time either because of bug fixing or new feature implementation. The IM-SCF binaries consist of two files with the extension of WAR:

* imscf-sl.war – for Signaling Layer
* imscf-el.war – for Execution Layer

The files should be placed into the directory /home/imscfadmin/imscf/imscf\_1\_0/imscf\_deployment on the oam01 machine. After copying invoke the distributeWars.sh script:

$ distributeWars.sh <domain>

for all affected domains, so replace the <domain> parameter above with “sless”, “sful” and “hlr”.

This command copies the new binaries to all the nodes and directories where the IM-SCF instances can find them. After distributing, the command resetAll.sh should be called on all affected machines.