

## Terminology

A **filter-engine** is a piece of code which can do certain type of filtering.

A **filter-config** is a set of parameters which configure a filter-engine. Filter-config are specific to the engine type while they share some common parameters.

A **filter-rule** is a combination of a filter-engine with a specific filter-config

A **filter-set** is a list of filter-rules in a specific order.

An **application-point** is the point in the message flow where the filters are being placed on. It can be on a specific inbound link or on a specific outbound link or in the routing phase.

A **filter-action** is a trigger which is telling the system what to do with this packet. Some filter-actions are exclusive (like route it to a specific output port) some could be applicable multiple times such as send a copy of the packet to a debug log A and to debug log B.

A **named-list** is a list of items with a name. Such a list can be used in multiple places for multiple different types.

For example you can make a list of specific numbers to monitor and refer to this list in multiple places through its name.

When a filter-set is put into action at an application-point, then the sequence of filter-rules is being processed and filter-actions are being collected and applied along the way.

The following section describe the parameters of each object. The objects will be manageable by a generic API where the parameters are named exactly the same way.

## SS7-Filter Staging Area

A **staging area** is a environment where modifications on to objects can be done while not being used.

A staging area can become a live config and a new staging area can be created. In terms of config files, this will be represented as individual config directories and a symbolic link to the active one.

As API calls

To create a new staging area

```
/api/ss7-filter-staging-area-add?name={name}
```

To delete a staging area

```
/api/ss7-filter-staging-area-delete?name={name}
```

To copy a staging area to another

```
/api/ss7-filter-staging-area-action?name={source-name}  
&action=copy  
&destination={destination-name}
```

To activate a current staging area

```
/api/ss7-filter-staging-area-action?name={name}  
&action=activate
```

To list current staging areas

```
/api/ss7-filter-staging-area-list
```

To select the staging area to edit engine status

```
/api/ss7-filter-staging-area-action?name={name}  
&action=select
```

Note: the selected staging area is specific to the current session.

## SS7 Filter Engine

In the config file:

```
group = ss7-filte-engine  
name = {object-name}  
filename = {filename}  
enable = {boolean}
```

As API calls

To add a new engine from a shared library file:

```
/api/ss7-filter-engine-add?name={name}  
&file={filename}
```

To remove an engine

```
/api/ss7-filter-engine-delete?name={name}
```

To activate an engine

```
/api/ss7-filter-engine-modify?name={name}&enable=YES
```

To deactivate an engine

```
/api/ss7-filter-engine-modify?name={name}&enable=NO
```

To list the current engine names

```
/api/ss7-filter-engine-list
```

To read the current engine config

```
/api/ss7-filter-engine-read?name={name}
```

To read the current engine status

```
/api/ss7-filter-engine-status?name={name}
```

To reload a new version of the engine file

```
/api/ss7-filter-engine-action?name={name}&action=reload
```

## SS7 Filter Ruleset

```
group = ss7-filte-ruleset  
name = {object-name}  
description = {text}  
status = { on | off | monitor }
```

As API calls

To add a new filtering ruleset

```
/api/filter-ruleset-add?name = {object-name}  
    &description={description}  
    &status={status}
```

To remove a filtering ruleset

```
/api/ss7-filter-ruleset-delete?name={name}
```

To activate a rule

```
/api/ss7-filter-ruleset-modify?name={name}  
    &status=on
```

*This switches all individual filtering rules to status on.*

To deactivate a rule

```
/api/ss7-filter-ruleset-modify?name={name}&status=off
```

*This switches all individual filtering rules to status pff.*

To put rule in monitoring mode

```
/api/ss7-filter-ruleset-modify?name={name}&status=monitor
```

*This switches all individual filtering rules to status pff.*

In monitor mode, all actions do take the monitor-action-list instead of the active-action-list. If status is set to off, the bypass-action-list is taken which defaults to continue.

To read the list of known rulesets in the config

```
/api/ss7-filter-ruleset-list
```

To read the current ruleset config

```
/api/ss7-filter-ruleset-read?name={name}
```

To read the current rule status

```
/api/filter-ruleset-status?name={name}
```

To send an action to the rule

```
/api/filter-ruleset-action?name={name}&action={action}
```

## SS7 Filter Rule

```
group = ss7-filte-rule
filter-ruleset = {object-name}
entry-nr = {integer}
description = {text}
status = { on | off | monitor }
engine = {name-of-filter-engine}
engine-config = {config}
filter-action-list = {action-list-name}
bypass-action-list = {action-list-name}
monitor-action-list = {action-list-name}
```

As API calls

To add a new filtering rule

```
/api/filter-rule-add?filter-ruleset = {object-name}  
    &entry-nr={integer}  
    &description={description}  
    &status={status}  
    &engine={name-of-engine}  
    &engine-config={config}  
    &bypass-action-list={action-list-name}  
    &filter-action-list={action-list-name}  
    &monitor-action-list={action-list-name}
```

If the conditions of the specific engine does match, then the actions are taken which are specified in the action list.

To remove a filtering rule

```
/api/ss7-filter-rule-delete?filter-ruleset={name}&entry-  
nr={integer}
```

To activate a rule

```
/api/ss7-filter-rule-modify?filter-ruleset={name}&entry-  
nr={integer}&status=on
```

To deactivate a rule

```
/api/ss7-filter-rule-modify?filter-ruleset={name}&entry-  
nr={integer}&status=off
```

To put rule in monitoring mode

```
/api/ss7-filter-rule-modify?filter-ruleset={name}&entry-  
nr={integer}  
    &status=monitor
```

In monitor mode, all actions do take the monitor-action-list instead of the active-action-list. If status is set to off, the bypass-action-list is taken which defaults to continue.

To read the list of known rules in the config

```
/api/ss7-filter-rule-list=filter-ruleset={name}
```

To read the current rules config

```
/api/ss7-filter-rule-read?filter-ruleset={name}&entry-nr={integer}
```

To read the current rule status

```
/api/filter-rule-status?filter-ruleset={name}&entry-nr={integer}
```

To send an action to the rule

```
/api/filter-rule-action?filter-ruleset={name}&entry-nr={integer}&action={action}
```

## SS7 Filter Action List

In the config file:

```
group = ss7-filte-action-list  
name = {object-name}  
a }  
log = {ss7-logfile-object-name}  
error = {errorcode}  
reroute-destination = {sccp-destination-name}  
reroute-called-address = {new-sccp-called-address}  
reroute-called-address-prefix = {new-sccp-called-address-prefix}  
category = {category-name}  
description = {text}
```

As API calls

To add a new filter-action

```
/api/ss7-filter-actionlist-add?name={name}  
&action={ "pass" | "drop" | "log" | "reroute" }  
&log={ss7-logfile-object-name}  
&reroute={sccp-destination-name}  
&category={category-name}
```

To remove a filter-action

```
/api/ss7-filter-actionlist-delete?name={name}
```

To list all filter-actions in the system

```
/api/ss7-filter-actionlist-list
```

To read the filter-action-config

```
/api/ss7-filter-actionlist-read?name={name}
```

To read the current engine status

```
/api/filter-actionlist-status?name={name}
```

To send an action to the filter action

```
/api/filter-actionlist-action?name={name}&action={action}
```

## SS7 Filter Action List

In the config file:



```

group = ss7-filte-action
name = {object-name}
action = { "pass" | "drop" | "abort" | "reject" | "error" |
"continue" | "log" | "reroute" | "set-category" | "clear-
category" }
log = {ss7-logfile-object-name}
error = {errorcode}
reroute-destination = {sccp-destination-name}
reroute-called-address = {new-sccp-called-address}
reroute-called-address-prefix = {new-sccp-called-address-
prefix}
category = {category-name}
description = {text}

```

As API calls

To add a new filter-action

```

/api/ss7-filter-actionlist-add?name={name}
&action={ "pass" | "drop" | "log" \ "reroute" }
&log={ss7-logfile-object-name}
&reroute={sccp-destination-name}
&category={category-name}

```

To remove a filter-action

```

/api/ss7-filter-actionlist-delete?name={name}

```

To list all filter-actions in the system

```

/api/ss7-filter-actionlist-list

```

To read the filter-action-config

```

/api/ss7-filter-actionlist-read?name={name}

```

To read the current engine status

```

/api/filter-actionlist-status?name={name}

```

To send an action to the filter action

```
/api/filter-actionlist-action?name={name}&action={action}
```

## Built in SS7 Filter Actions

There are several predefined actions.

pass

the filtering ruleset processing is stopped and the packet is forwarded normally

drop

the filtering ruleset processing is stopped and the packet is dropped.

abort

the filtering ruleset processing is stopped and a tcap abort message is sent back.

reject

the filtering ruleset processing is stopped and a tcap end with a gsmmap reject message is sent back.

error(errorcode)

the filtering ruleset processing is stopped and a tcap end with a gsmmap ReturnError message is sent back with an indicated error

continue

the filtering ruleset is continued on the next filtering rule.

log(logfile)

the packet is logged to the indicated logfile object.

```
reroute(destination,new-called-address)
```

the packet is rerouted to another destination and the called address is replaced if set.

```
add-tag(tag)
clear-tag(tag)
clear-all-tags
```

the tag is set or cleared. Note a packet can have multiple tags. clear-all-tags would clear them all;

```
set-custom(customvar,value)
```

sets a custom field to a specific value.

## SS7 Log File

In the config file:

```
group = ss7-log-file
name = {object-name}
file = {filename}
format = { pcap | text }
rotate-minutes = {minutes}
rotate-packets = {packets-count}
enable = {boolean}
```

As API calls

To add a new log file object

```
/api/ss7-log-file-add?name={name}
&file={filename}
&format={"text" | "pcap"}
&rotate-minutes={minutes}
&rotate-packets={packets-count}
&enable={boolean}
```

To start logging

```
/api/ss7-log-file-modify?name={name}&enable=YES
```

To stop logging

```
/api/ss7-log-file-modify?name={name}&enable=N0
```

To remove a log-file-object

```
/api/ss7-log-file-remove?name={name}
```

To list all log-file objects

```
/api/ss7-log-file-list
```

To read the current log-file-object config

```
/api/ss7-log-file-read?name={name}
```

To send an action to the log-file-object

```
/api/ss7-log-file-action?name={name}&action={action}
```

Action can be “reopen”

## SS7 Filter engine “screening”

The configuration of the screening engine is a json dictionary with the following objects

```
tag = {tag1,tag2,...}
custom1 = {value}
custom2 = {value}
custom3 = {value}
custom4 = {value}
custom5 = {value}
custom6 = {value}
custom7 = {value}
custom8 = {value}
custom9 = {value}
custom10 = {value}

gsm-map-operation-code = {int}
gsm-map-operation-code-list = {named-list}

application-context = {int}
application-context-list = {named-list}

application-context-version = {int}
application-context-version-list = {named-list}

sccp-calling-address-ton = {integer}
sccp-calling-address-npi = {integer}
sccp-calling-address-number-list = {namedlist-name}
sccp-calling-address-prefix-list = {namedlist-name}

sccp-called-address-ton = {integer}
sccp-called-address-npi = {integer}
sccp-called-address-number-list = {namedlist-name}
sccp-called-address-prefix-list = {namedlist-name}

originating-pointcode-list = {namedlist-name}
destination-pointcode-list = {namedlist-name}

imsi-prefix-list = {namedlist-name}
```

The tag is a element which can be set by other filters. If it is set and not set to “any” then the filter would only apply if the tag matches.

If a tag name is set with a ! in front, this means the tag must not

be present. Individual tags are separated by a comma and all the conditions listed in the tag statement must apply.

custom1 .. custom10 are similar field which can be used in results.

If gsm-map-operation-code is set, then this rule only applies to packets with this operation code. Otherwise it applies to all. Operation codes can also be negated by having a “!” in front of them.

If application-context is set, then this rule only applies to packets with this application-context. Otherwise it applies to all. It can also be negated by having “!” in front of the entries.

## SS7 Filter engine “sms”

The configuration of the sms engine is a json dictionary with the following fields

```
tag = {tag1,tag2,...}
custom1 = {value}
custom2 = {value}
custom3 = {value}
custom4 = {value}
custom5 = {value}
custom6 = {value}
custom7 = {value}
custom8 = {value}
custom9 = {value}
custom10 = {value}

gsm-map-operation-code = {int}
gsm-map-operation-code-list = {named-list}

application-context = {int}
application-context-list = {named-list}

application-context-version = {int}
application-context-version-list = {named-list}

sms-type = {MO or MT}
```

```
application-context = {named-list-name}  
sms-sender-ton = {int}  
sms-sender-npi = {int}  
sms-sender-number-list = {named-list}  
sms-sender-prefix-list = {named-list}  
sms-receiver-ton = {int}  
sms-receiver-npi = {int}  
sms-receiver-number-list = {named-list}  
sms-receiver-prefix-list = {named-list}  
sm-sc-ton = {int}  
sm-sc-npi = {int}  
sm-sc-number = {number}  
sm-sc-number-list = {named-list}  
sm-sc-prefix = {number-prefix}  
sm-sc-prefix-list = {named-list}
```

The category is a element which can be set by other filters. If it is set and not set to “any” then the filter would only apply if the category matches.

If gsm-map-operation-code is set, then this rule only applies to packets with this operation code. Otherwise it applies to all. Operation codes can also be negated by having a “!” in front of them.

If application-context is set, then this rule only applies to packets with this application-context. Otherwise it applies to all. It can also be negated by having “!” in front of the entries.

sccp-calling-address-blacklist-action is the action taken if the sccp-calling-address-blacklist test is failing. The

The sccp-calling-address-blacklist will call the action “drop” or the action if the SCCP calling address is found in the list with the indicated name. This requires an exact match.

The sccp-calling-prefix-blacklist will call the action “drop” if a prefix is found in the named list which is matching the beginning of the SCCP calling address.

## SS7 Filter engine “location”

This filter engine will check the location

Location process could be enhanced by adding some checks at the

reception of a new LocationUpdate (or UpdateGPRSLocation).

“Check Location” could be implemented in various ways:

- Compare with the previous location: analyse if it is physically possible to move from the previous country to this new one
- Check the status on the previous VLR by sending, for example, a PSI message to the old VLR and check that the user is no longer active on this old VLR before sending the CancelLocation
- Update the VLR address in the HLR database before sending the LocationUpdate Ack or at TCAP-End
- Correlate the LocationUpdate (or UpdateGPRSLocation) with MAP messages generated earlier in order to verify that VLRid is not faked, like a successful authentication on the same MSC / VLR or SGSN / VLR (SendAuthenticationInfo).

The configuration of this module is still under review. This is a stateful filter which is a tricky one.

## SS7 Filter engine “cap”

This filter engine will check CAMEL messages

It compares current VLR and calling SCCP address

It is a stateful message as it has to remember the VLR (from previous UpdateLocation)

The configuration of this module is still under review.