ZephyrTel Mobile Messaging FAF R02.12.05

Operator Manual

Release 18.12 Revision A

January 2021

ZephyrTel

CloudForward

Table of Contents

List of Figures

Chapter 1

Introduction

Topics:

1.1 About this Document

This document contains all relevant details required for the operation and administration of the ZephyrTel Mobile Messaging Firewall Advanced Filter (FAF).

The FAF is a product from the ZephyrTel Mobile Messaging product family of SS7 message routing and network querying products.

Since the available functions are licensed and depend on the specific implementation, not all functions and/or applications contained in this document may be relevant or applicable to the system you will be working with. Actual screen presentation may differ from the screens presented in this document due to software changes or browser configurations.

The FAF can be used in combination with a Router (RTR) or a Firewall (FWL). For consistency, this document will only use the term FWL. Anything that is said about the FWL also applies to the RTR.

1.2 Scope

This document discusses the functionality of the ZephyrTel Mobile Messaging FAF component.

1.3 Intended Audience

This document is meant for everybody interested in how the FAF can best be used, but mainly for:

- Implementation Engineers who are responsible for the pre-installation, on-site installation and configuration of the FAF in the end-user environment.
- Maintenance and Support Engineers who are responsible for maintaining the total system environment of which the FAF is a part.
- Network Operators who are in charge of the daily operation of the ZephyrTel Mobile Messaging systems and infrastructure.

1.4 Documentation Conventions

Typeface or Symbol	Meaning	Example
Bold	Refers to part of a graphical user interface.	Click **Cancel**.
`Courier`	Refers to a directory name, file name, command, or output.	The `billing` directory contains...
`<pointed brackets>`	Serves as a placeholder for text that the user will replace, as appropriate in context.	The file is called `MGRdata.xml.<ip>.gz`, where `<ip>` is the server's IP address.

Typeface or Symbol	Meaning	Example
`[square brackets]`	Indicates an optional command.	`[--validateonly]`
Note:	Indicates information alongside normal text, requiring extra attention.	**Note:** Ensure that the configuration...
\ (Unix)	Denotes line continuation; the character should be ignored as the user types the example, and ENTER should only be pressed after the last line.	`% grep searchkey \` `data/*.dat`

1.5 Locate Product Documentation on the Customer Support Site

Access to ZephyrTel's Customer Support site is restricted to current ZephyrTel customers only. This section describes how to log into the ZephyrTel Customer Support site and locate a document. Viewing the document requires Adobe Acrobat Reader, which can be downloaded at www.adobe.com.

1. Log into the ZephyrTel Customer Support site.

   Note: If you have not registered for this new site, click the **Register Here** link. Have your customer number available. The response time for registration requests is 24 to 48 hours.

2. Click the **Product Support** tab.
3. Use the Search field to locate a document by its part number, release number, document name, or document type. The Search field accepts both full and partial entries.
4. Click a subject folder to browse through a list of related files.
5. To download a file to your location, right-click the file name and select **Save Target As**.

Chapter 2

System Overview

Topics:

2.1 Introduction

This chapter describes the role of the FAF in the context of a complete ZephyrTel Mobile Messaging system.

A system involving the FAF in combination with a Firewall (FWL) can be configured to implement advanced filtering of SMS messages. Such a filter is typically used to reject SPAM messages, modify offensive messages or alert the network operator of sudden increases of certain SMS-related network traffic.

Additionally, the FAF can also be used in combination with the Subscriber Provisioning Framework (SPF) to automatically blacklist Originator or Recipient subscribers sending/receiving SPAM or fraudulent messages.

Furthermore, it is possible to use FAF in combination with the MGR to automatically blacklist a specific GT or Network.

2.2 System Context

The picture below shows the use of the FAF in a typical system.

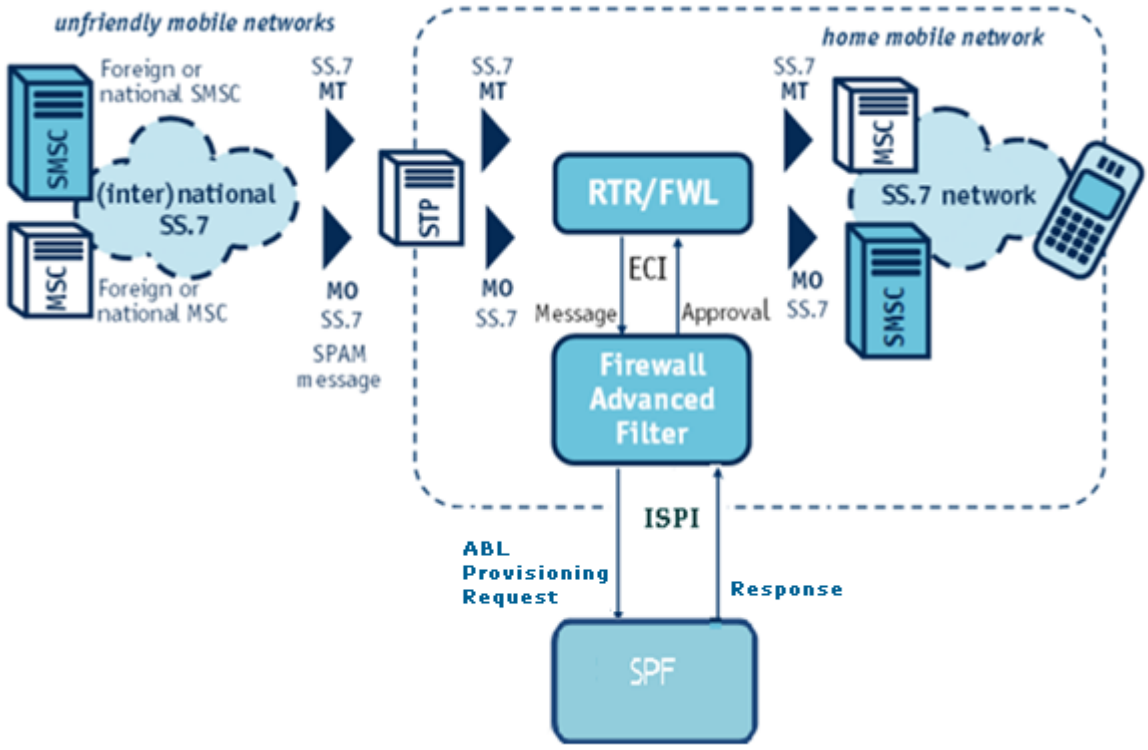


Figure 1: FAF System context

The Signalling Transfer Points (STPs) of the home mobile network are configured to route mobile-originated (MO) and mobile-terminated (MT) messages from a foreign, possibly unfriendly network to the FWL for advanced filtering.

The FWL uses the External Condition Interface (ECI) to provide the FAF with the relevant details of received MO and MT messages.

The FAF applies its configurable logic and returns a Boolean value (true or false) which the FWL can use to pass or block the message. By default, a result of true means “approved” or “pass”, while a result of false means “not approved” or “block”. However, the FWL can be configured to handle the results differently. If the FWL decides to pass a message, the message is released into the friendly home mobile network.

Additionally, if Automatic Blacklisting (ABL) parameters are configured on certain FAF filters having appropriate conditions to match a spam or fraudulent message, then the FAF sends auto blacklist provisioning requests to the SPF for blacklisting the corresponding originating or recipient subscribers. The FWL as usual decides to discard messages received from blacklisted originators or destined to blacklisted recipients based on the ECI return result from the FAF, which is typically “false” in such cases.

2.3 System Components

The FAF does not have an interface to communicate directly with non-ZephyrTel network nodes; it must always be combined with the Router/Firewall.

The ECI protocol is used for all communication between the Firewall (FWL) and the FAF. The ECI protocol runs on top of a TCP/IP network stack.

Although logically, the FAF provides a service to the FWL, for network security reasons the ECI protocol is designed so that the FAF acts as the client connecting to the server on the Firewall.

For the Automatic Blacklisting (ABL) functionality, the FAF supports the Internal Service Provisioning Interface (ISPI) with the SPF. The ISPI protocol runs on top of a TCP/IP network stack. The FAF acts as the client connecting to the server on the SPF.

2.4 System Configuration

The FAF is configured over SNMP. Configuration can be done in the following ways:

- File-based configuration of static node parameters in an XML configuration file that is processed during FAF start-up.
- GUI-based configuration of the filter logic of the FAF. The Manager (MGR) provides a Web-based GUI for provisioning most filter logic parameters that the FAF provides.

Note: The memory requirements of the FAF strongly depend on which filter conditions are configured. For example, the volume conditions may require a large amount of memory depending on traffic rates, tracking period, and condition type. In case of doubt if your current system has enough memory available, please contact ZephyrTel for correct memory requirements.

2.5 Firewall Configuration

To provide the FAF with the relevant message parameters:

- The Firewall (FWL) must be configured to recognise the FAF as an EC application
- External condition rules must be created to apply FAF-based filtering

When the FWL receives a message, it evaluates the external condition rules that apply to that type of message (MO, MT, AO or AT). If a matched rule specifies that the FAF plays the role of the external condition, the message parameters enabled for the external condition are sent to the FAF over ECI.

The FWL processes the Boolean result that the FAF returns to decide what action should be taken for the message. Possible actions include:

- Pass the message as-is
- Pass the message after modification
- Reject the message
- Block the message, but return an acknowledgement to the message originator

2.6 Redundancy

Redundancy can be provided by configuring more than one FAF to connect to the same Firewall (FWL) node. This method ensures that, if a FAF node is unavailable, the FWL can distribute its requests over the remaining FAF nodes.

Also, a FAF node can connect to multiple FWL nodes, so if a FWL node is unavailable, the FAF can continue to communicate with other FWL nodes.

For more information about redundancy, refer to the Router (RTR) Operator Manual.

Note: When configuring the FWL, only one instance of the EC application is required to connect with multiple instances of the FAF that fulfil the same purpose. However, their internal states will differ, so they will not behave exactly the same at all times.

For the Automatic Blacklisting (ABL) functionality, redundancy can be provided on the FAF - SPF interface by configuring multiple FAFs to connect to the same SPF node. This would ensure that even if a FAF node is unavailable, still auto blacklist provisioning requests can be sent to the SPF through the remaining FAF nodes.

Also, a single FAF can connect to multiple SPF nodes, such that if one SPF is unavailable the FAF can continue to communicate with other SPF nodes for auto blacklist provisioning.

2.7 Multi-Instance Support

Multi instance feature allows multiple ZMM users (up to 10, including the existing `textpass` user) be created on the same node, each of whom will be able to run one instance of FAF.

Note: A separate LICENSE is required for each ZMM user.

Chapter 3

FAF Overview

Topics:

3.1 Introduction

The FAF can detect and modify SMS messages and alert the operator when specific messages are processed. This chapter provides an overview of the FAF interfaces and architecture.

3.2 Interfaces

3.2.1 External Condition Interface

The FAF receives message fields from the RTR/FWL and returns its filter logic over the external condition interface (ECI) protocol, which runs on top of the TCP/IP network stack. The FAF requires a user name and password to log on to the FWL. A single FAF can connect to multiple FWL nodes.

In a multi-instance setup, if multiple instances of FAF are supported on the same node, then each FAF would also require the user id (i.e. id assigned by operating system to the user under whose context the FAF instance is running) along with the user's login id and password while sending the ECI login request; the default user id is 200, corresponding to the UID of the `textpass` user.

Refer to *Configuration* for information about establishing a connection between the FAF and the FWL.

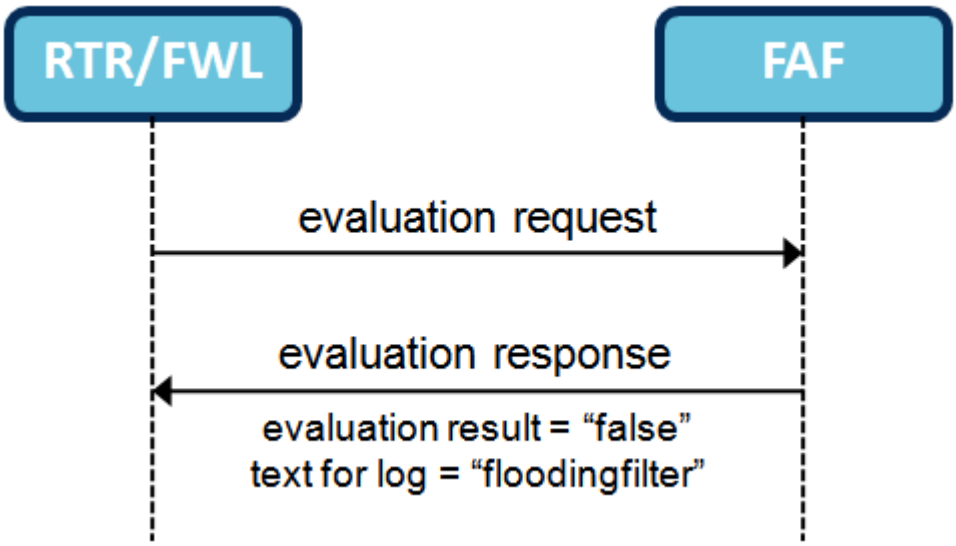


Figure 2: ECI interaction

For example, the figure demonstrates the interaction between the FAF and the RTR/FWL. The FWL sends an evaluation request message to the FAF to start the filter logic. The evaluation request message should contains all information that is required by the filter logic, such as originator and recipient number, message header, and user data.

In this example, the filter is matched and flooding is detected. The FAF then indicates to the FWL to reject the message by setting the evaluation result field in the evaluation response message to "false". In addition, the matched filter name “floodingfilter” is also returned to the FWL for logging purposes.

The FAF can have a maximum of **100** ECI connections to the RTR/FWL.

Note: The number configured ECI connections will not have any influence on the total throughput of the FAF

3.2.2 Internal Service Provisioning Interface (ISPI)

If the Automatic Blacklisting (ABL) functionality is enabled and configured on the FAF, it sets up an Internal Service Provisioning Interface (ISPI) connection with the SPF in order to exchange auto blacklist provisioning request and response messages. The ISPI protocol runs on the top of the TCP/IP network stack, and the FAF acts as a client and logs on to the SPF server immediately after a connection is created.

Each FAF supports only one ISPI connection with a single SPF node. However, a single FAF can connect to multiple SPF nodes over separate connections. Refer to *Configuration* for more information about establishing a connection between the FAF and the SPF.

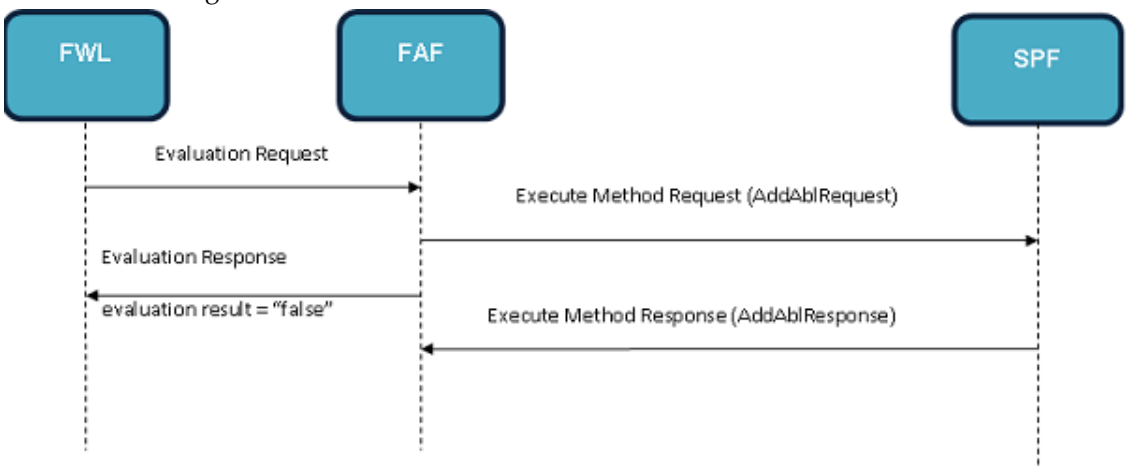


Figure 3: ISPI Interaction

The above figure demonstrates an example of the interaction between the FAF and the SPF. The FAF sends an Execute Method Request message to the SPF to add a subscriber to the ABL provisioning information maintained in the SPF database. The Execute Method Request message should contain all the information that is required by the SPF for automatically blacklisting a subscriber, including the subscriber MSISDN and the blacklist duration. The SPF sends back an Execute Method Response message indicating the success or failure of the auto blacklist provisioning operation.

In the above example, once a filter having ABL parameters configured matches the message received from the RTR/FWL, the FAF indicates to the RTR/FWL to reject the message by setting the evaluation result field in the Evaluation Response message to "false". In addition, the FAF also indicates to the SPF to blacklist the originator/recipient subscriber of the message for the configured duration (or permanently).

The FAF can support a maximum of **25** ISPI connections, with an equal number of SPF nodes.

Note: The number of configured ISPI connections will not have any influence on the total throughput of the FAF.

3.2.3 SNMP

Provisioning is done over SNMP. Refer to *OAM Interface (SNMP)* for information about SNMP-based provisioning.

3.3 Message Fields

The FAF operates on the message fields that the FWL provides. It is recommended to configure the RTR/FWL to only provide the message fields on which the FAF operates. Each of the FAF's condition types operates on a single message field. Generally, the use of one specific field is recommended for each condition type. The following message fields are generally recommended for any condition type:

Message Field	Description
Data	The user data portion of a message.
Header	The user data header of a message, as described in 3GPP TS 23.040 v6.2.0.
Smsc	The address of the SMSC that issued the MT SMS.
Orig	The address of the message originator.

Note: For Automatic Blacklisting purposes it is mandatory to include the MSISDN of the message originator/recipient in the ECI Request from the RTR/FWL. Any other form of subscriber address, including alphanumeric addresses, are currently not supported for the ABL functionality.

3.4 Automatic Blacklisting

3.4.1 Overview

The Automatic Blacklisting functionality is implemented by enhancing the ZMM anti-fraud system. As part of this enhanced functionality, if a MO or MT message is detected as “spam” or “fraudulent” by appropriately configured FAF filters, the subscriber who had sent the message or for whom it was destined would automatically get blacklisted in the Subscriber Provisioning Framework (SPF) database. Once a subscriber is blacklisted, all subsequent messages sent by or destined for this subscriber would be rejected by the RTR/FWL, either for a configurable time period or permanently. If the subscriber is blacklisted for a specific time period, then upon expiry of that period the subscriber will be removed from the blacklist in the subscriber provisioning database and he or she would be able to send messages to or receive messages from the SMS network again.

Important: A small time-interval is likely to elapse after the expiry of a subscriber’s blacklist period before that subscriber is actually removed from the blacklist and his/her messages are allowed to pass by the RTR. This small time-interval is not expected to exceed 5 minutes under normal operational conditions, provided only one Auto Blacklist Service (see below) is provisioned in the system.

The RTR/FWL decides to send an incoming MO/MT message to the FAF based on whether or not an ABL service is already active for the originating/recipient subscriber, as indicated by the Service

Subscription Information (SSI) component, and also depending on the configuration of the relevant External Condition (EC) Rules on the RTR. Refer to the RTR Operator Manual for more details.

Once a FAF filter with suitably configured conditions for detecting spam or fraudulent messages matches a MO/MT message, if ABL-specific parameters are also configured on the same filter then the FAF sends a provisioning request to the SPF to automatically blacklist the corresponding originating or recipient subscriber for a specific time-duration or permanently, as per the ABL configuration settings.

The following diagram depicts the automatic blacklisting operation in the overall context of a MO/MT message flow:

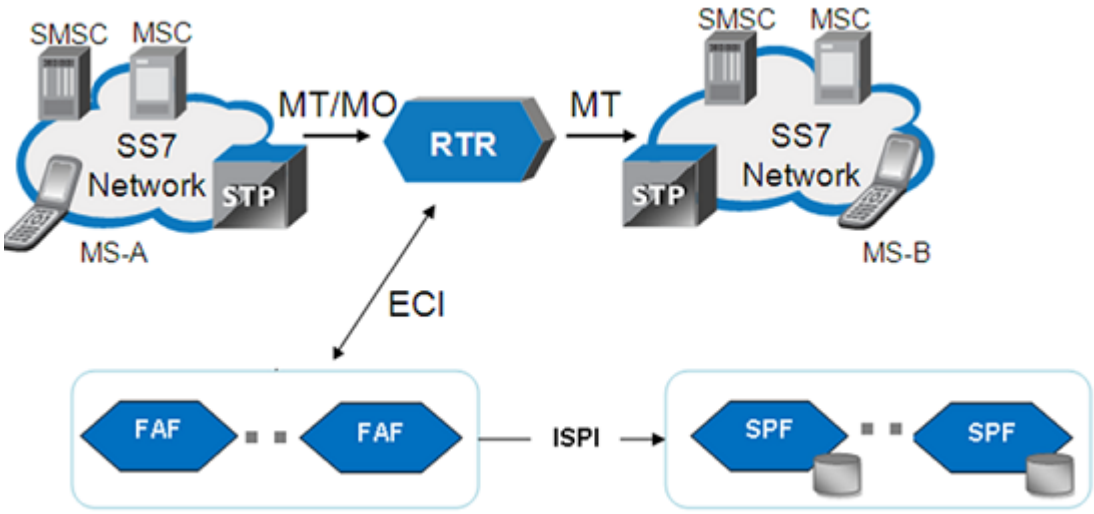


Figure 4: ABL operational context

If the FAF is connected to more than one SPF node over ISPI, it load-shares the auto blacklist provisioning requests among the ISPI connections using a round-robin policy. The FAF also maintains pre-defined limits on the maximum allowed number of pending requests on an ISPI connection (i.e. for which no responses have been received yet from the SPF) and the maximum time-duration it would wait for receiving a response from the SPF for a particular request (before a timeout occurs).

For the ABL functionality, the maximum allowed number of pending requests per ISPI connection is **500**, and the maximum time-duration to wait before a SPF response timeout occurs is **5 seconds**. These values are not configurable.

Important: The FAF is capable of supporting up to **200** auto blacklist provisioning requests per second, irrespective of the number of SPF nodes it is connected to.

3.4.2 Auto Blacklist Service

A new service type 'Auto Blacklist' has been introduced in order to allow the operator to provision ABL-specific services (from the MGR GUI) for automatic blacklisting of originator and recipient subscribers. It is a special service category which is neither a Personalized service nor a Value-added Non Provisionable Service (NPS) but the operator will still have full control regarding service activation and service parameter configuration (e.g. the blacklist duration) for each subscriber to be blacklisted.

- Originator ABL service refers to a service type that is used to automatically blacklist subscribers who send messages that are detected as spam or fraudulent by FAF filters.

- Recipient ABL service refers to a service type that is used to automatically blacklist subscribers who are the intended recipients of messages detected as spam or fraudulent by FAF filters.
- If the profile of a subscriber, identified by MSISDN, for example, already exists in the SPF database, and the profile does not have ABL enabled, then the ABL provisioning for this subscriber will fail.

Refer to the SPF Operator Manual and the RTR Operator Manual for more details.

3.5 Automatic GT or Network Blacklisting

This functionality is implemented based on MGR lists of type MSISDN and their usage in the routing rule conditions. In other words, blocking of blacklisted parties will be achieved by adding them in a list of type MSISDN, which is used by a routing rule condition.

When the filter matches, and the blacklist type is either Originating GT or Originating Network, then FAF sends the blacklist party (either GT or network as chosen when creating the filter) to the MGR via the existing XML/HTTP interface, and it is added to configured ABL list. The list that is updated with the blacklisted entry (or entries in case of network) would be used by the RTR via a routing rule to block traffic.

Note: For SSN Routing, the RTR uses the home network as originating network. In this case, the home network may be blocked if the user selects to block Originating Network option while configuring the filter.

Note: The RTR will not block the incoming MO traffic when the ABL List is configured in **Orig. MSC/SGSN[cond]** of a MOR Rule if the routing is based on PC/SSN.

The following figure captures the high-level design:

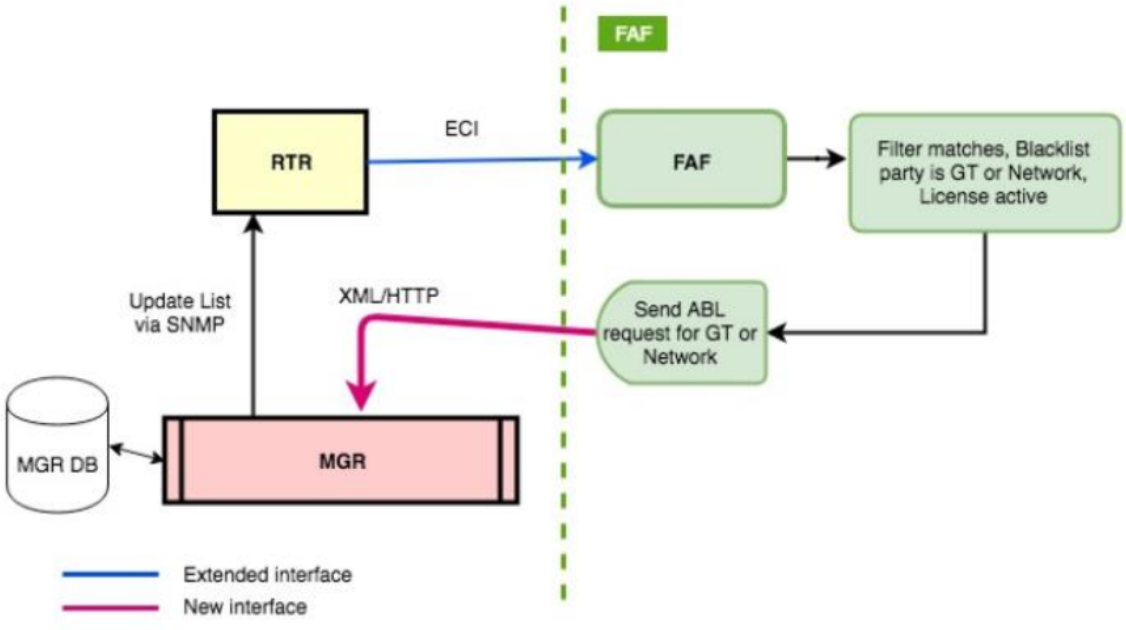


Figure 5: Flow Diagram

Note: For MT-ST-MT and MT-ST-AT, any blacklisting should be performed on MTIX.

3.6 Filter Logic

The FAF blocks/passes messages according to the configured filters and actions. When the FAF is configured in "monitoring" mode, the "Action" of filter must be set to "true". This way the RTR/FWL will *not* block the message; it will only log it for future analysis. When FAF is configured to "filter" traffic, the action of filter must be set to "false". This way the RTR/FWL will block the message and create a log for it.

Additionally, a FAF filter can be configured to automatically blacklist the originator or intended recipient of a message, either permanently or for a specified duration of time.

3.6.1 Terminology

- Filter Logic—A prioritised set of filters.
- Filter—The FAF supports up to a combination of 100 filters. Each filter has an associated priority and action:
  - The priority is a number between 0 and 100.

    Note: The priority between -100 to 0 shall no longer be used. The priority shall be rearranged to [0, 100] range if negative priority is used.

  - A filter consists of a chain of filter conditions.
  - If the filter is matched (all conditions are satisfied), the action is applied.
  - The action determines whether the FAF returns a result (true or false) to the FWL, or continues the filter logic.
  - In case FAF returns a result of *true* based on filter matching and action setting, an additional option can be used to append text (such as "SPAM") to the end of the message.
  - Additionally, if the filter is matched against a MO/MT message, then the blacklist party is checked.
  - The blacklist party determines whether the FAF should send auto blacklist provisioning requests to SPF for the Originator or Recipient subscriber of the matching MO/MT message. If the blacklist party is set to "None", it indicates that the filter is not configured to support ABL functionality.
  - If the blacklist party is not "None", the blacklist action determines whether subscriber should be blacklisted permanently or temporarily (for the blacklist duration).
- Condition Type—Each condition type implements a specific algorithm that uses message fields as its input and returns a Boolean result (true or false). A condition type can be instantiated multiple times in a filter or filter logic. For a detailed description of the condition types, refer to *Advanced Filtering*.
- Filter Condition—Each filter supports up to 100 filter conditions. Each filter condition instantiates a condition type. A filter condition can be configured to invert (change from "matching condition" to "not matching condition") the result of the condition type instance before passing it to the filter processing. If the invert is not set (default), the condition returns true when it is matched, and false otherwise. If the invert is set, the condition returns false when it is matched.

3.6.2 Filter Architecture

The FAF processes messages by evaluating the provisioned filter logic. You can provision up to 100 filters. Each filter contains one or more filter conditions. FAF evaluates filters according to their priority, beginning with the highest priority filter.

The figure illustrates the FAF logic.

- FILTER 99 is the filter with priority 99
- Condition 99-1 is the first condition of filter 99
- Type A/B/C is the type of the condition

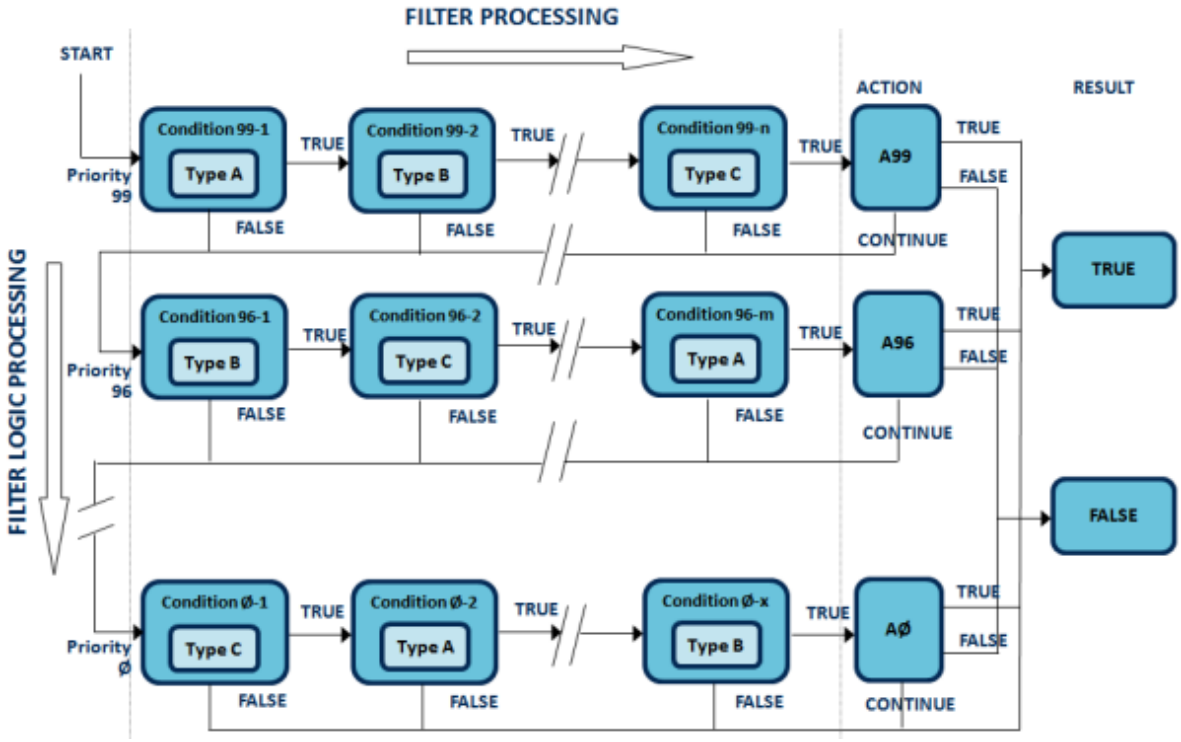


Figure 6: FAF filter processing

FAF is designed to support complex Boolean functions. In most cases, the implemented filter logic will consist of no more than five filter conditions, arranged in simple logic. However, to successfully design the desired filter logic, it is important to understand the logic structure of FAF.

3.6.2.1 Filter Priority

If any of the filter's conditions returns true, it means that the message matches the filter. In this case, the logic is finished, unless the filter's action is "continue". The filters with lower priority are not processed for this message if the logic is finished. In addition, the FAF returns the filter's name to the FWL (if the name is not empty).

If any of the filter's conditions returns false, it which means that the filter is not matched. Evaluation continues at the first condition of the next (lower priority) filter. If no lower priority filter is present, the filter logic defaults to a final result of true.

3.6.2.2 Filter Action

If the filter is matched, the filter's action is applied to the message. The filter's action is one of the following:

- Return True—The FAF returns true for the message fields
- Return False—The FAF returns false for the message fields
- Continue—The FAF should continue to process the next filter

Note: An empty filter (chain) without any condition (block) will be always matched. Therefore, do not activate a filter without any condition. **Before** activating a filter, please activate first all the required conditions. In case a filter is deactivated, make sure that all the active conditions are deactivated **after** the filter is deactivated.

3.6.2.3 Filter Blacklist Party

If the filter is matched and ABL license is enabled, the filter's Blacklist Party decides whether the Originator or the Recipient of the message should be blacklisted. The blacklist party can be set to one of the following:

- None: No auto blacklisting request is sent by the FAF for this matching filter.
- Originator: The FAF sends auto blacklist request for the message Originator.
- Recipient: The FAF sends auto blacklist request for the message Recipient.

If the "Auto GT Network Blacklist" license is enabled, there will be two more options for the blacklist party as follows:

- Originating GT - The FAF sends auto blacklist provisioning request for the GT of the SMSC.
- Originating Network - The FAF sends auto blacklist provisioning request for the Network of the SMSC.

3.6.2.4 Filter Blacklist Service

This field will be available only when the Filter Blacklist Party is selected as "Originator" or "Recipient".

The filter's Blacklist Service specifies the particular Originator ABL Service or Recipient ABL Service, using which the subscriber is to be blacklisted in the SPF.

3.6.2.5 Filter List

This field will be available only when the Filter Blacklist Party is selected as "Originating GT" or "Originating Network".

The filter's List specifies the ABL List which the SMSC GT or Network Number Range will be added to.

The list to be used can be created from **Routing ➤ Others ➤ List** tab.

3.6.2.6 Filter Blacklist Action

The filter's Blacklist Action specifies the nature of auto blacklisting supported by this filter.

The blacklist action can be set to one of the following:

- Permanent Blocking: Subscriber is to be blacklisted permanently.

- Time-based Blocking: Subscriber is to be blacklisted for a specified time period.
- Absolute Blocking: SMSC GT or Network is to be blacklisted until a specific date time. This option is available only when the Filter Blacklist Party is selected as "Originating GT" or "Originating Network".

3.6.2.7 Filter Blacklist Duration

The filter's Blacklist Duration specifies the time-duration for which subscriber should be blacklisted. This parameter is relevant only if the filter's Blacklist Action is set to "Time-based Blocking".

3.6.2.8 Filter End Date

The filter's End Date specifies the last date time for which subscriber or network should be blacklisted.

This parameter is relevant only if the filter's Blacklist Action is set to "Absolute Blocking".

3.6.2.9 Filter Conditions

If a condition returns true, the next condition of the same filter is evaluated. If all conditions in the filter return true, the filter is matched. The filter's action is applied.

If a condition returns false, the filter is not matched. The next filters are processed according to their priority.

If a filter does not have any conditions, FAF considers the filter to be matched.

3.7 Condition Types

The FAF contains the following condition types:

- Content—Detects a word or phrase from a provisioned list
- Duplicates—Detects messages that are very similar in a relatively large number of recent messages
- Flooding—Detects sudden increases in traffic from the same originator(s)
- Volume—Detects and counts messages sent by a single originator and/or with a specific content within a certain time span
- Bulk—Detects multiple messages sent by a single originator within a certain time span
- Enhanced messaging (EMS)—Detects messages containing specific Information Element IDs (IEI) in the user data header provided by the FWL
- Expression—Evaluates an expression containing message fields

Note: Each condition type is licensed separately.

Refer to *Advanced Filtering* for more information.

3.8 User Data Modification

CAUTION: Do not use text replacement/append functions that may make the user data longer than the original user data. When applied to a "full segment" this will lead to an undeliverable message.

If the filter is matched, and the action is not *continue*, the text provided in the **Append** field is appended to the message data (if the data message field has been provided to the FAF).

The content filter has its own modification mechanism. Refer to *Advanced Filtering* for more information.

Note: To enable the FAF to modify the message text, the corresponding external condition (EC) application must be configured to allow modification of the message.

3.9 Lists

The FAF's content condition matches words or phrases to entries in a provisioned list. For example, a list can contain banned words or spam message templates. Lists are independent of the content conditions; multiple content conditions can refer to the same list, if necessary.

The FAF supports up to 100 lists, and each list can contain up to 1000 entries. The semi-static configuration attribute `maxlistlength` controls the maximum internal memory units that the FAF can use for each list. Its range is 6400 to 64,000 bytes (the default is 6400). For example, if maxlistlength is set to 6400, then in a list with 100 entries, each entry can be 64 bytes. If the list had 1000 entries, each entry could be 6.4 bytes.

This functionality allows you to optimise the FAF's memory usage, depending on the content filtering that you want to do. For example, if your list of sensitive words is less than 6400 bytes, setting `maxlistlength` to 6400 will allow the FAF to allocate enough internal memory units for your list, without allocating unneeded memory units that will impact performance. If you provision a list that is too large for the allocated memory to hold it, the FAF will write a warning message in the syslog.

Refer to *Advanced Filtering* for more information about the content condition type.

3.10 Provisioning

3.10.1 Provision a Filter

The Manager (MGR) interface is used to provision the FAF filter logic. An overview of the provisioned filters appears in **Advanced Filters ➤ Filters**

The status icon indicates the filter's state: active or inactive. To view or edit a filter, click its row. To deactivate, delete, copy, or re-sync a filter, select the box in its row and select the appropriate action from the **Action** menu.

To add a new filter, click **Add New**. Give the new filter a name and, optionally, a description. Set the new condition's priority between 0 (lowest priority) and 100 (highest priority).

Note: Each filter must have a unique priority. Assigning a new filter the same priority as an existing one will be rejected by the MGR.

The priority between -100 to 0 shall no longer be used. Rearrange the priority to the [0, 100] range if negative priority is used.

Additionally, you may configure the parameters related to automatic blacklisting on a filter. Refer to *Dynamic Configuration* for details.

3.10.2 Provision a Condition

Select a filter, and click on **Add New** under the **Filter Conditions**. Optionally provide a description of the condition. From the **Type** list, select the condition type (content, flooding, EMS, or duplicates). From the **Error Condition** list, select the specific condition to use. The detail provisioning of each filter type is interpreted in chapter *Advanced Filtering* .

Select **Invert** if you want to let the condition return *false* when it is matched, and *true* when it is not matched. For instance, a white list is provisioned as invert of all allowed addresses. If the address is not in the white list, the condition returns *true*, and the message is further evaluated.

3.10.3 Provision a List

Provision lists using the Manager interface. An overview of the provisioned lists appears in **Advanced Filters ➤ Lists**.

The status icon indicates the list's state: *active* or *inactive*. To view or edit a list, click its row. To deactivate, delete, copy, or re-sync a list, select the box in its row and select the appropriate action from the **Action** menu.

To add a new list, click **Add New**. Give the new list a name and, optionally, a description.

A list consists of multiple entries, which can be single words or phrases. Each list entry must begin on a new line in the **Text** box; the FAF does not support multi-line entries.

3.11 Logging

When a FAF filters matches and the filter action is "return false" or "return true", the matching FAF filter name is send to the Router (RTR) in the ECI evaluation response message for logging purposes. This allows operators to analyze logged messages more effectively, and tune the FAF filters if needed.

The log field `textInEvaluationResponse` indicates the FAF filter name in the Router log records.

Example

An incoming MO message is detected to be flooding by the FAF. The FAF sets the `evaluationResult` to false and sends the name of the filter that was matched (for example "flooding filter") in the ECI evaluation response message to the RTR. The RTR generates the log with the `textInEvaluationResponse` indicating the filter name "flooding filter". The RTR rejects the short message by sending a negative acknowledgement to the MSC.

Chapter 4

Advanced Filtering

Topics:

4.1 Introduction

This chapter explains the Firewall Advanced Filter (FAF) condition types.

Before the explanation of each condition type, the text pre-processing is introduced. This is because the text pre-processing is used by the **Content** and the **Duplicates** condition types.

4.2 Pre-Processing Message Content

Before FAF compares a message to the provisioned content and duplicates conditions, FAF pre-processes the content (user data) of the message and any filter lists that are used with the content conditions. Pre-processing reduces the resources that FAF's algorithms require to evaluate each message by reducing the amount of information contained in the message.

Pre-processing consists of three consecutive steps:

Name	Description	Input	Output	Used By
Tokenisation	Map similar characters to strings of "tokens" that you define in the normalisation table	The message content or an entry in a content condition list	A string of tokens that FAF generates based on the normalisation table	The content condition when its accuracy is set to "tokenise"
Normalisation	Collapse consecutive identical tokens into a single token	The output of the tokenisation step	A normalised string of tokens	The content condition when its accuracy is set to "normalise"
Featurisation	Group four consecutive tokens into a "feature"	The output of the normalisation step	The "signature" of the message	The duplicates condition

4.2.1 Tokenisation

During tokenisation, similar characters are mapped to the same token. The characters that are mapped to each token can be set using the semi-static configuration attribute `normalisationmap`.

Tokenisation automatically removes all of the following:

- Any character that is not in the mapping table
- Any character that is mapped to 0 (zero)
- White space

4.2.1.1 Format of the Tokenisation Map

The format of the `normalisationmap` attribute is a list of character groups separated by the new line character (`
`).

All characters of the first character group map to token 1, all characters of the second character group map to token 2, and so on. All characters that are not specified in the map are dropped during tokenisation.

Specify international characters using the decimal HTML encoding of the Unicode character. Refer to *http://www.utf8-chartable.de/unicode-utf8-table.pl?unicodeinhtml=dec* for character codes.

For example, assume that:

- Characters `a`, `A`, and a-umlaut (`ä`) should be mapped to token 1
- Characters `b`, `B`, and `6` should be mapped to token 2
- Character `C` should be mapped to token 3

The character code for a-umlaut is 228. Therefore, the configuration file should contain:

```
normalisationmap="aA&#228;&#10;bB6&#10;C"
```

Note: Changes to the tokenisation mapping will overwrite the FAF's default mapping. If you want to append to the default mapping table, ensure that you preserve it when you modify the `normalisationmap` attribute.

To view the current tokenisation mapping on a FAF system, execute:

```
tp_walk fafPropertiesNormalise
```

4.2.1.2 Default Tokenisation Map

The FAF's default mapping of characters to tokens is:

```
normalisationmap="0oO&#246;&#214;&#10;1iIlL!\/&#10;2zZ&#10;3eE&#10;4aA&#228;&#196;
&#10;5sS&#223;&#10;6&#10;7tT&#10;8bB&#10;9gG&#10;cC&#10;dD&#10;fF&#10;hH&#10;
jJ&#10;kK&#10;mM&#10;nN&#10;pP&#10;Q&#10;rR&#10;uU&#252;&#220;&#10;vV&#10;
wW&#10;xX&#10;yY"
```

This translates to:

Token	Characters
1	0, o, O, ö, Ö
2	1, i, I, l, L, !, \, /
3	2, z, Z
4	3, e, E
5	4, a, A, ä, Ä
6	5, s, S, $, ß
7	6
8	7, t, T
9	8, b, B
10	9, g, G
11	c, C
12	d, D
13	f, F

Token	Characters
14	h, H
15	j, J
16	k, K
17	m, M
18	n, N
19	p, P
20	q, Q
21	r, R
22	u, U, ü, Ü
23	v, V
24	w, W
25	x, X
26	y, Y

4.2.1.3 Sample Tokenisation Maps

A sample tokenisation map that includes French characters is:

```
normalisationmap="0oO&#246;&#214;&#244;&#212;&#10;1iIlL!\/&#238;&#206;&#239;
&#207;&#10;2zZ&#10;3eE&#233;&#201;&#232;&#200;&#234;&#202;&#235;&#203;
&#10;4aA&#228;&#196;&#226;&#194;&#10;5sS&#223;&#10;6&#10;7tT&#10;8bB&#10;
9gG&#10;cC&#231;&#199;&#10;dD&#10;fF&#10;hH&#10;jJ&#10;kK&#10;mM&#10;nN&#10;
pP&#10;qQ&#10;rR&#10;uU&#252;&#220;&#250;&#218;&#251;&#219;&#10;vV&#10;wW&#10;
xX&#10;yY&#255;&#159;"
```

A sample tokenisation map that includes Arabic characters is:

```
normalisationmap="0oO&#246;&#214;&#1569;&#10;1iIlL!\/&#1575;&#1573;
&#1571;&#1570;&#10;2zZ&#1572;&#10;3eE&#1574;&#10;4aA&#228;&#196;&#1576;&#10;
5sS&#223;&#1578;&#1577;&#10;6&#1579;&#10;7tT&#1580;&#10;8bB&#1581;&#10;9gG&#1582;
&#10;cC&#1583;&#10;dD&#1584;&#10;fF&#1585;&#10;hH&#1586;&#10;jJ&#1587;&#10;
kK&#1588;&#10;mM&#1589;&#10;nN&#1590;&#10;pP&#1591;&#10;qQ&#1592;&#10;rR&#1593;
&#10;uU&#252;&#220;&#1594;&#10;vV&#1601;&#10;wW&#1633;&#1602;&#10;xX&#1634;
&#1603;&#10;yY&#1635;&#1604;"
```

A sample tokenisation map that includes Russian (Cyrillic) characters is:

```
normalisationmap="&#x410;&#x430;&#10;0Oo&#246;&#214;&#10;6&#x411;&#x431;
&#10;1IiLl!\/&#10;&#x412;&#x432;&#10;2Zz&#10;&#x413;&#x433;&#10;&#x414;&#x434;
4Aa&#228;&#196;&#10;3Ee&#x415;&#x435;&#x417;&#x437;&#10;5Ss&#223;&#10;&#x416;
&#x436;6&#10;7Tt&#10;&#x418;&#x438;8Bb&#10;&#x419;&#x439;9Gg&#10;&#x41a;&#x43a;
Cc&#10;&#x41b;&#x43b;Dd&#10;&#x41c;&#x43c;Ff&#10;&#x41d;&#x43d;Hh&#10;&#x41e;
&#x43e;Jj&#10;&#x41f;&#x43f;Kk&#10;&#x420;&#x440;Mm&#10;&#x421;&#x441;Nn&#10;
&#x422;&#x442;Pp&#10;&#x423;&#x443;Qq&#10;&#x424;&#x444;Rr&#10;&#x425;&#x445;
Uu&#252;&#220;&#10;&#x426;&#x446;Vv&#10;&#x427;&#x447;Ww&#10;&#x428;&#x448;
Xx&#10;&#x429;&#x449;Yy&#10;&#x42a;&#x44a;&#10;&#x42b;&#x44b;&#10;&#x42c;
&#x44c;&#10;&#x42d;&#x44d;&#10;&#x42e;&#x44e;&#10;&#x42f;&#x44f;"
```

4.2.1.4 Tokenisation Examples

These examples of tokenisation are based on the default tokenisation map:

1. The string `many dollars` is tokenised into:

   ```
   17,5,18,26,12,1,2,2,5,21,6
   ```

   After tokenisation, the following strings are equal:

   ```
   many dollars
   M4NyD011Ar5
   ```

2. The string `E l l e n` is tokenised into:

   ```
   4,2,2,4,18
   ```

   After tokenisation (and removal of white spaces), the following strings are equal:

   ```
   E l l e n
   Ellen
    E llen
   ```

4.2.2 Normalisation

After tokenisation, FAF applies normalisation, which collapses multiple identical tokens into a single token.

4.2.2.1 Normalisation Examples

Some examples:

1. In the first example in *Tokenisation*, two token 2s appear consecutively (representing the two letter Ls in dollar). During normalisation, they are collapsed into a single token 2, so the string becomes:

   ```
   17,5,18,26,12,1,2,5,21,6
   ```

   After normalisation, the following strings are equal:

   ```
   many dollars
   maany dolar$s
   ```

2. In the second example in *Tokenisation*, the normalised result is:

   ```
   4, 2, 4, 18
   ```

   The following strings are all mapped to the same token:

   ```
   elen
   elllen
   e llen
    e l l e n n
   ```

4.2.3 Featurisation

After normalisation, the FAF applies featurisation to the normalised string of tokens to create features. A feature is a combination of four consecutive tokens.

During featurisation, the FAF translates a string of tokens into a set of features by considering each substring of four tokens as one feature. A feature can occur multiple times in a string.

While the order of tokens in a string is important, a set of features has no "order". Therefore, featurisation helps the FAF's algorithms match text fragments, independent of their order.

4.2.3.1 Featurisation Example

The example string `"many dollars"` is tokenised into:

```
17,5,18,26,12,1,2,2,5,21,6
```

Then, the tokens are featurised as follows:

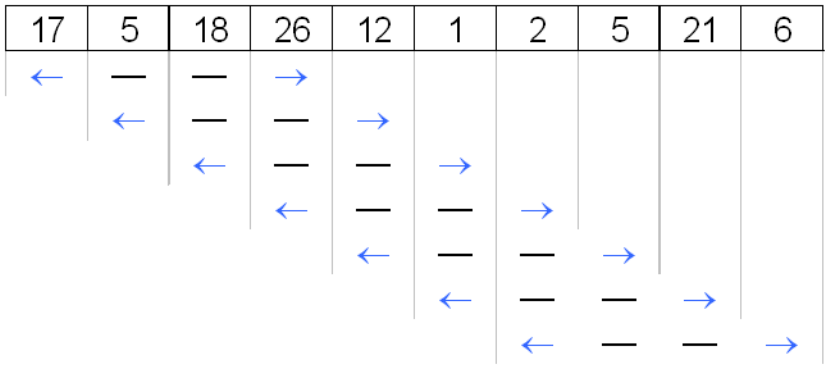


Figure 7: Featurisation example

The example string contains seven features, which are in an arbitrary order:

```
17-5-18-26
2-5-21-6
26-12-1-2
1-2-5-21
12-1-2-5
5-18-26-12
18-26-12-1
```

4.3 Content Filtering

The FAF's content condition detects whether the content of a configured message field matches a word, phrase, or regular expression in a provisioned list. If the content of the field matches at least one of the items in the list, the condition returns "true" for the message being evaluated. The accuracy of word or phrase detection is configurable, ranging from an exact match to a match after normalisation.

The content condition is most commonly used to evaluate the content of the user data (message content) field.

The FAF supports up to **100** instances of the content condition.

Note: In the FAF MIB and license file, the content filter is referred to as the string filter.

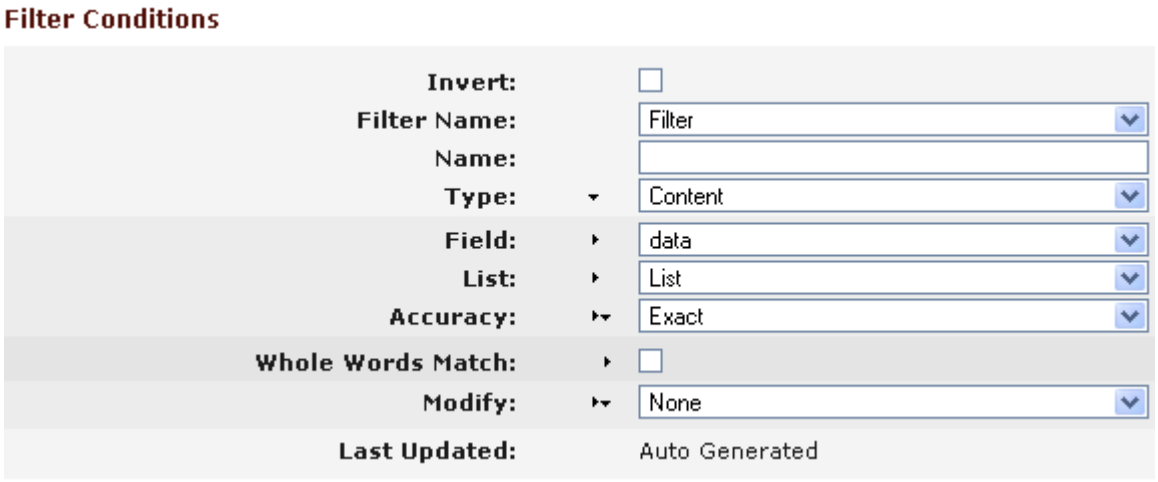


Figure 8: Content condition MGR configuration

4.3.1 Content Condition Whole Word Matching

The whole words match option performs matching of whole words. For example:

- Exact matching: `"apple"` matches text `"this is an apple."`, but not `"this is a pineapple."`
- Case-insensitive matching: `"apple"` matches `"Apple is good."`, but not `"PineApple is good."`

The whole words matching option applies to Exact and Case-insensitive matching only.

Word

A word is a combination of characters, which are **not** in the word boundary set. Refer to *Word Boundary Rules* for recognition of a word.

Word Boundary

The word boundary can be any UTF-8 character that is defined as word boundary by the configuration. The following characters are considered word boundaries in most languages and by default, the FAF will consider these as word boundary:

- Start of text (SOT) and End of text (EOT);
- White space;
- Carrier return (CR) and Line Feed (LF);
- New line character: New Line (NEL);
- Control characters of ASCII and extended ASCII set (`0x00,0x1f` and `0x7f,0x9f`];
- Other not-printable ASCII characters;
- Punctuation used in most Latin languages as word breaks:
  `~`!@#$%^&*()+-_=[]{}\|:";'<>?,./`

It is possible to define a customized word boundary set using *wordboundary*. When not provisioned, the default word boundary set will be used.

Word Boundary Rules

The FAF uses the following rules to determine a word:

- The word must be immediately after a word boundary;
- The word must not contain any word boundary;
- The word must be immediately followed by a word boundary;
- By default A_B is considered two words A and B. This is controlled by the word boundary set. If the "_" is **not** in the customized set, A_B is considered one word.

4.3.2 Modification of User Data

FAF can modify the User Data when a match is identified and the "modify" field is not "none".

There are three types of modification:

1. maskString: The complete matched string is replaced by the "mask". The length of the message is not changed if the mask and the original string has equal length. The length is reduced when the mask is shorter than the original string.
2. replaceString: The matched string is replaced by the "replace string". The length of the string can be changed if the replacement is not equal to the original string. If the length of the modified string is longer than the maximum length of a data segment, the string is truncated to the maximum length of a data segment (140 bytes using 7-bits encoding, for example).
3. replaceMessage: The whole message is replaced by the "replace string". The length of the message is changed to the length of the "replace string".

Note: The "replace string" cannot be longer than the maximum length of a data segment (140 bytes in 7-bits encoding, for example).

4.3.3 Content Condition Examples

Detect Combinations of Words

You can use two or more content conditions to detect combinations of words. For example, assume you want to detect and block messages that contain the words "free" and "money". However, you do not want to block messages that contain only "free" or only "money". To accomplish this:

1. Ensure the EC application for the FAF is configured to send the data message field to the FAF
2. Create two lists: one containing "free", the other containing "money"
3. Create two filters
4. Open one filter and add a content condition
5. For the **Field** parameter, select data
6. For the **List** parameter, select one of the lists that you created
7. Open the other filter and add a content condition
8. For the **List** parameter, select the other list that you created
9. Ensure that the filter with the higher priority has an **Action** of "continue"

If both filters return "true", then both words are present in the message.

Detect Numeric Originators

To use the content condition to detect numeric originators:

1. Ensure the EC application for the FAF is configured to send the originator address message field to the FAF
2. To detect numeric originators with international numbers, create a list that contains the following regular expression:

   ```
   ^[+][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]$
   ```

   To detect numeric originators with national numbers, create a list that contains the following regular expression:

   ```
   ^[N|U][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]$
   ```

3. Create a filter
4. Add content condition to the filter
5. For the **Field** parameter, select "originator address"
6. For the **List** parameter, select the list that you created
7. For the **Accuracy** parameter, select "regular expression"

Detect Short Code Originators

To use the content condition to detect short code originators:

1. Ensure the EC application for the FAF is configured to send the "originator address" message field to the FAF
2. Create a list that contains the following regular expression:

   ```
   ^[N|U][1-9][0-9][0-9][0-9][0-9][0-9]$
   ```

3. Create a filter
4. Add a content condition to the filter
5. For the **Field** parameter, select "originator application"
6. For the **List** parameter, select the list that you created
7. For the **Accuracy** parameter, select "regular expression"

Detect Protocol ID field

To use the content condition to detect the message having configured protocol id.

Assume you want to detect messages that contain protocol id as 50. To accomplish this:

1. Ensure the EC application for the FAF is configured to send the message field (protocolId) to the FAF.
2. Create list with either a single specific value (50) or a set of possible values including the desired value (50) of protocol ID.
3. Create one filter ‘F’.
4. Open filter ‘F’ and add a content condition.
5. For the **Field** parameter, select “protocolId”.
6. For the **List** parameter, select one of the lists that you created.

7. For the **Accuracy** parameter, select "Exact" and for the **Modify** parameter, select "None".
8. Select the **Whole Words Match** option.

4.3.4 Content Condition Required Message Fields

The message field that you specify in the content condition's **Field** parameter must be sent to the FAF. To ensure that it is, select this message field in the EC application that you create for the FAF.

4.3.5 Content Condition Traps

The duplicates condition may issue the following SNMP traps:

- `stringListExceedMaxLengthAlert`: the list used by this content filter exceeds the maximum list length. The `maxlistlength` of the FAF property shall be increase. The default value of this parameter is 6400 bytes.

Refer to the ZephyrTel Mobile Messaging SNMP Trap Reference Guide for more information about these traps.

4.4 Duplicates Filtering

The FAF's duplicates condition detects messages that are very similar to a relatively large number of recent messages. Positive detection returns true.

The duplicates condition measures similarity by comparing a number of features (see *Pre-Processing Message Content*). If the similarity is set to 100%, exact matching is performed.

The duplicates condition performs cluster detection, through which it attempts to detect groups of similar messages that are large enough to be worthy of tracking. If it detects such a group, the duplicates condition creates a cluster for this group of messages and starts to track them accurately (called cluster matching).

The FAF supports up to **10** instances of the duplicates condition.

Note: A parameter `enablealternativeduplogic` controls the algorithm to calculate the similarity of the featurised text. Please refer to the *enablealternativeduplogic* for detailed explanation.

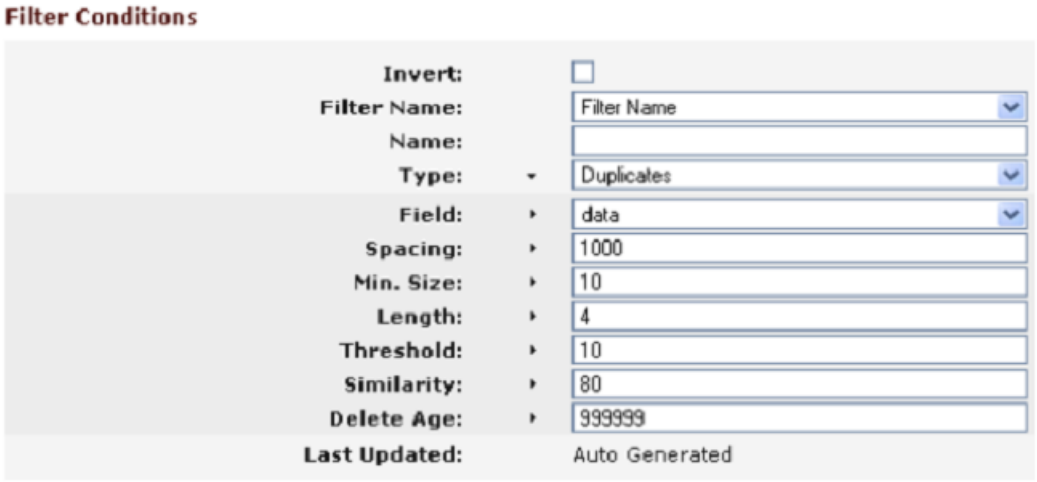


Figure 9: Duplicates condition MGR configuration

4.4.1 Detecting Duplicates Clusters

The duplicate condition considers the entire history of features that are recorded. A cluster is created when enough features in the message have been recorded in the history and the appearance frequency exceeds the configuration.

The creation of clusters is related to the configuration, the traffic load, and the traffic pattern. The configuration impacts clusters as follows:

- **Similarity**—The lower the similarity, the more clusters can be created. The similarity is calculated based on how many features in a new incoming message were previously seen in a different message. For example, assume the new incoming message contains 10 features, and 8 of the 10 features are found in the old message; this means the similarity is 8/10 = 80%.

  Note: If the similarity is set to 100%, the **Length** parameter will be ignored and it will detect duplicate messages with any length.

- **Length**—Total number of features in the message. If the message is short, less features can be generated from the message. The length should be big enough to avoid matching commonly used short messages like `"Hello!"`.
- **Minimum Size**—The minimum number of the SIMILAR messages required before creating a cluster. If (similar msg_cnt > min_size), a new cluster is created. Then the newly incoming messages will be first matched against the clusters.

  Example: In total 4 similar messages will trigger a cluster based on min_size=2

  - the 1st message has NO duplicate, the counter is 0
  - the 2nd message makes the similar counter 1
  - the 3rd makes the counter 2
  - the 4th makes the counter 3, which is bigger than 2, so a cluster is created.

- **Spacing**—Number of messages between similar messages. If more messages than the spacing are found between two messages, the messages are not considered duplicates. For example, if a user sends `"hello"` twice and there are many other messages in between, it is not a duplicate attack.

Set this field to a value that is big enough if your traffic is high and a duplicate attack is likely to happen.

- **Delete Age** (duration of a cluster being stored in the FAF)—The longer the delete age, the more clusters can be created.

The more messages there are, the more patterns can be found; consequently, there will be more clusters. Some common patterns, such as very frequently used words like `"where"`, increase the chance of cluster creation.

4.4.2 Duplicates Cluster Matching

The process of matching duplicates clusters is:

1. When the duplicates condition evaluates a message, it compares the message to all existing clusters.
2. If the message matches one cluster, the process is finished and the condition returns "true".
3. If the message does not match any cluster, the condition compares the message against the feature pool and the message history.
4. The condition adds the message and its features to the history.
5. If a match is found or if the number of duplicates has not yet reached the limit, the process is finished and the condition returns "false".
6. If a match is found and the number of duplicates has reached the limit, the condition creates a new cluster, the process is finished, and the condition returns "true".

Clusters expire (that is, are forgotten) if no new duplicates of the messages in the cluster are found during a configurable period of time.

4.4.2.1 Duplicates Cluster Alerts

When the number of clusters that the FAF is currently tracking exceeds a configurable threshold (by default, 300), the FAF will issue the `clustersInUseAlert` SNMP trap. The trap will indicate the number of clusters being tracked and the warning level:

- `warningThreshold`—The number of clusters is above the configured threshold
- `limitationReached`—The number of clusters has reached the maximum that the FAF supports (500)

The FAF issues the `clustersInUseAlertClean` trap when the number of clusters is 50 less than it was when the `clustersInUseAlert` trap was issued.

To check the number of clusters being tracked at any given time, execute the following at a command prompt on a FAF system:

```
tp_walk fafPropertiesDupsClustersInUse
```

To check whether the number of clusters is below or above the configured threshold, or at the maximum limit, by executing:

```
tp_walk fafPropertiesDupsClusterStatus
```

4.4.3 Limitations of Duplicates Cluster Tracking

The FAF stores historic information using memory and evaluates the history using CPU. Therefore, to prevent the FAF from using too much memory and CPU, there is a 500-cluster limitation. The more clusters that are created, the slower the FAF becomes and the more memory and CPU it will use.

Each duplicates condition can track up to 65,536 messages at one time. Each new message will overwrite an old message in this count.

The FAF determines message similarity by processing each message into features (refer to *Pre-Processing Message Content*), which are stored in the FAF's feature pool. The feature pool only remembers the message ID of the last match; therefore, if more than one message contains the same feature, the duplicates condition only remembers the last message that contained that feature. In some cases, this may prevent duplicates matching.

4.4.3.1 Troubleshooting Duplicates Cluster Tracking

There are several considerations to note when you are troubleshooting the duplicates condition.

One is that when MSCs receive duplicate messages from a single originator, they load share the traffic over STPs, which in turn load share the traffic over the RTRs that communicate with individual FAF servers in the Mobile Messaging system. This load sharing enlarges the average time between each duplicate message, and can therefore cause the FAF to not recognize duplicates because, by the time a potentially matching message reaches the FAF, the FAF has already overwritten the message history of the oldest matching message.

Another consideration is that because the duplicates condition only remembers the last message that contained a given feature, duplicates may not match because other messages in between them have updated the feature history. For example, duplicate messages D1 and D2, and non-duplicate message N1, all contain feature F1. If the messages arrive in the order D1, N1, D2, then D2 will not match D1 because the last instance of F1 that the FAF remembers was in N1.

4.4.3.2 Duplicates Condition and Inaccuracy

The duplicates condition is a statistical condition and therefore introduces a certain degree of inaccuracy that may rarely cause the following behaviour:

- Detection of duplicates that are not exact matches (unless a similarity of 100% is configured)
- Delayed detection of a cluster due to overlapping features of non-duplicate messages between duplicated messages
- Failure to detect exact duplicates because the cluster is started with a similar, but not identical, message (the duplicate message will not cluster-match)

4.4.4 Duplicates Condition Required Message Fields

For the duplicates condition, the external condition (EC) application must be configured to send the following message fields to the FAF:

- Originator address
- Originator IMSI
- User data
- SMSC address
- MSC address

4.4.5 Duplicates Condition Traps

The duplicates condition may issue the following SNMP traps:

- `clusterStarted`—A new cluster was started/created.

- `clusterThreshold`—A cluster has grown beyond the configured size threshold; depending on the configuration, further duplicate messages may be blocked.
- `clusterExpired`—A cluster expired (the cluster is not matched for longer than the delete age).
- `clustersInUseAlert`—The number of clusters that the FAF is tracking (`fafPropertiesDupsClustersInUse`) crossed the configured threshold (`dupsclustertrapwarningthreshold` in the semi-static configuration file).
- `clustersInUseAlertClean`—The number of clusters that the FAF is tracking is 50 less than when the `clustersInUseAlert` was issued.

Refer to the ZephyrTel Mobile Messaging SNMP Trap Reference Guide for more information about these traps.

4.4.6 Troubleshooting the Duplicates Condition

Duplicates conditions can increase the complexity of the FAF's internal state. To investigate an instance of the duplicates condition, monitor the SNMP table `FAFDupsCountTable`. Consult the FAF MIB for a description of the columns of this table.

The following table describes some symptoms that a duplicates condition instance can show and some possible remedies.

Symptom	Action
No cluster created.	Reduce the **Similarity** value and/or increase the **Spacing** value.
Cluster created too late.	Reduce the **Min. Size** value.
Cluster created, but no action is taken.	Reduce the **Threshold** value.
Cluster expires too quickly.	Increase the **Delete Age** value.
Too many clusters.	• Increase the **Similarity** value. • Increase the **Length** value. • Increase the **Min. Size** value. • Reduce the **Delete Age** value. • Reduce the **Spacing** value.

4.5 Flooding Filtering

The FAF's flooding condition detects sudden increases in traffic from the same originator(s). Positive detection returns the result "true".

The flooding condition continuously monitors the short-term and long-term traffic averages (in messages per second) per originator (or range of originators). If the short-term traffic average exceeds the long-term traffic average by a configured margin for a configured period of time, flooding is detected and the condition returns "true".

The flooding condition remains in effect until the short-term traffic average drops below the level at which flooding was initially detected. While the flooding condition is in effect, the FAF does not update the long-term traffic average.

The FAF supports up to **10** instances of the flooding condition.

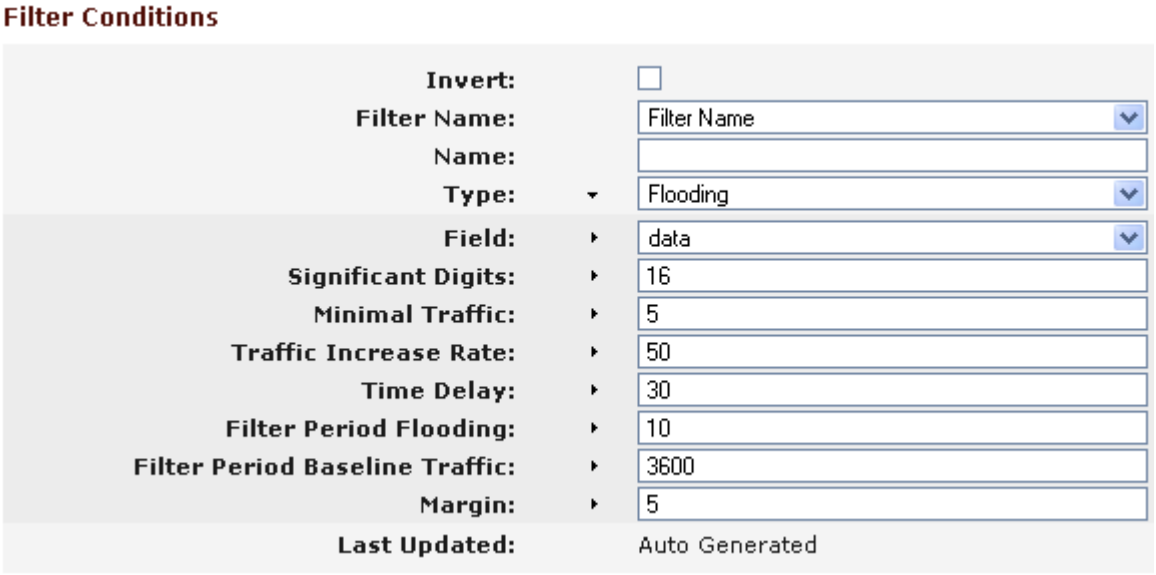


Figure 10: Flooding condition MGR configuration

4.5.1 Detecting Flooding

The short-term traffic average (`stta`) and long-term traffic average (`ltta`) are calculated as follows:

```
stta = number of messages received in the last X seconds / X seconds
ltta = number of messages received in the last Y seconds / Y seconds
```

Where `X` is the value of the **Filter Period Flooding** parameter and `Y` is the value of the **Filter Period Baseline Traffic** parameter.

The FAF detects flooding if the following condition evaluates to true for the configured period of time:

```
stta > ltta * (1 + rate/100) + threshold
```

Where `rate` is the value of the **Traffic Increase Rate** parameter and `threshold` is the value of the **Minimal Traffic** parameter.

The short-term and long-term traffic averages are first-order conditions; both have a configurable response time.

The FAF creates a tracking record for each encountered group of significant digits (**Significant Digits** parameter), if the record does not already exist. A maximum of 10,000 simultaneous tracking records can exist at one time. Every second, the FAF checks each tracked group to evaluate if both the long-term and short-term traffic are below margin/100 messages per second. If they are, the FAF deletes the tracking record for that number group.

Note: To prevent spurious flooding detection after the FAF is started or restarted, the FAF disables flooding detection for a time period (the long-term filter range plus the short-term filter range) after the first message has been received. This method allows the condition parameters to settle on stable values.

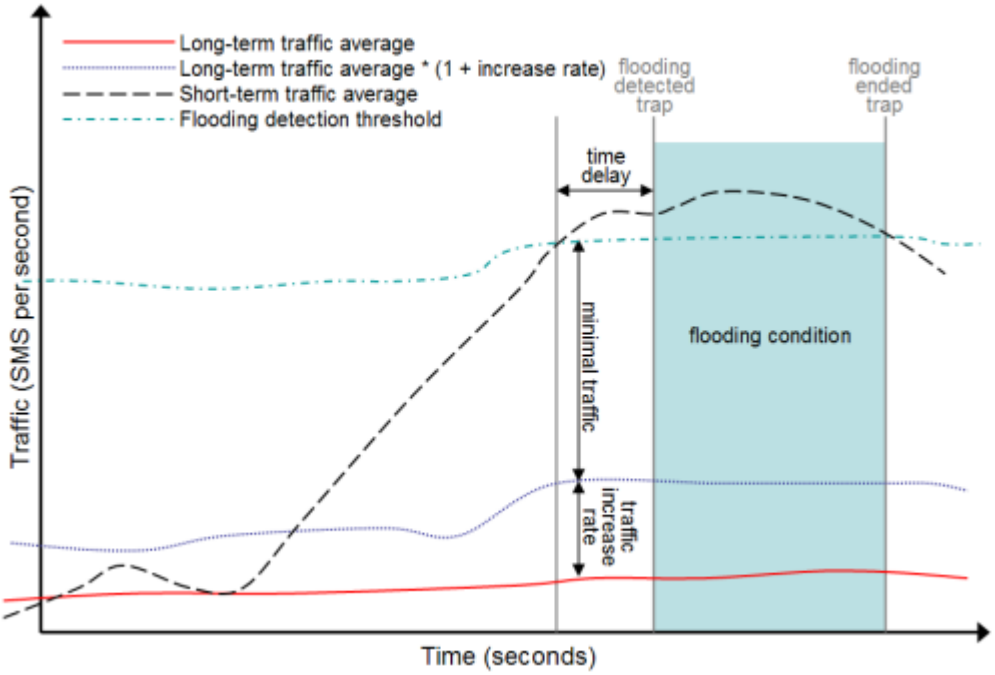


Figure 11: Flooding detection

4.5.2 Flooding Condition Example

In this example, the FAF flooding condition parameters are set as follows:

- Type—Flooding
- Field—orig
- Significant Digits—10
- Minimal Traffic—5 messages per second
- Traffic Increase Rate—100%
- Time Delay—5 seconds
- Filter Period Flooding—10 seconds
- Filter Period Baseline Traffic—120 seconds
- Margin—5 messages per 1000 seconds

A single originator sends 1 message per second for 150 seconds. Then, the originator starts sending 10 messages per second.

Until the 150th second afterward, the short-term and long-term traffic average are both 1 message per second. Flooding is not detected because the resulting condition is false:

```
1 > 1 * (1 + 100 / 100) + 5
```

At the 155th second:

```
stta = (5 * 1 + 5 * 10) / 10 = 5.5
ltta = (115 * 1 + 5 * 10) / 120 = 1.375
```

Flooding is still not detected because the resulting condition is false:

```
5.5 > 1.375 * (1 + 100 / 100) + 5
```

At the 158th second:

```
stta = (2 * 1 + 8 * 10) / 10 = 8.2
ltta = (112 * 1 + 8 * 10) / 120 = 1.6
```

Flooding is detected at this second because the resulting condition is true:

```
8.2 > 1.6 * (1 + 100 / 100) + 5
```

FAF waits 5 seconds (because the time delay is set to 5); if the the flooding condition persists, then FAF starts blocking all messages from the originator.

FAF continues to calculate the short-term traffic average; when it falls below 8.2, the flooding condition becomes false and traffic from the originator is allowed again. The long-term traffic average (1.6) remains constant while the flooding condition is true.

4.5.3 Flooding Condition Required Message Fields

For the flooding condition, the external condition (EC) application must be configured to send the message field specified in the flooding condition's **Field** parameter to the FAF.

4.5.4 Flooding Condition Traps

The flooding condition may issue the following SNMP traps:

- `floodingStartDetected`—Traffic from a foreign network has increased significantly.
- `floodingEndDetected`—Traffic from a foreign network has normalised (clears the `floodingStartDetected` trap).

Refer to the ZephyrTel Mobile Messaging SNMP Trap Reference Guide for more information about these traps.

4.5.5 Troubleshooting the Flooding Condition

Flooding conditions can increase the complexity of the FAF's internal state. To investigate an instance of the flooding condition, monitor the SNMP `FAFFloodCountTable`. Consult the FAF MIB for a description of the columns of this table.

The following table describes some symptoms that a flooding condition instance can show and some possible remedies.

Symptom	Action
Flooding detection is too sensitive.	Increase the **Traffic Increase Rate**.
Flooding detection is not sensitive enough.	Reduce the **Traffic Increase Rate**.
Flooding detection is too sensitive to short term traffic fluctuations.	Increase the **Filter Period Flooding** or increase the **Time Delay**.
Flooding detection is too slow.	Reduce the **Time Delay**.
Flooding detection is not sensitive enough to slow traffic increases.	Increase the **Filter Period Baseline Traffic** or reduce the **Traffic Increase Rate**.
Flooding detection affects unrelated originators.	Increase the **Significant Digits** or try a different message **Field**.

Symptom	Action
Flooding detection is too sensitive during low traffic periods.	Increase the **Minimal Traffic**.

4.6 Volume Filtering

The FAF's volume condition allows blocking of short messages (SMs) with certain characteristics, if their number exceeds a certain threshold in a limited period of time.

For example, the operator can use the volume filter condition to:

- Block traffic exceeding 50 SMS/day from the same originator
- Block traffic exceeding 30 SMS/12 hours where both the originator and message content are the same
- Block traffic exceeding 50 SMS/6 hours where the message content is the same.

The volume condition detects and counts SMs with the same *key fields* over a configurable tracking period. All SMs with the same key fields are counted in a *counting group*. A counting group is automatically created when a SM with a new set of key fields hits the condition. When a new SM with the same key fields hits the condition, the counter of the counting group is incremented. A certain period after the creation of the counting group, the counter is reset and the group is cleaned up. Whenever the counter exceeds a certain threshold, the volume condition returns 'true'.

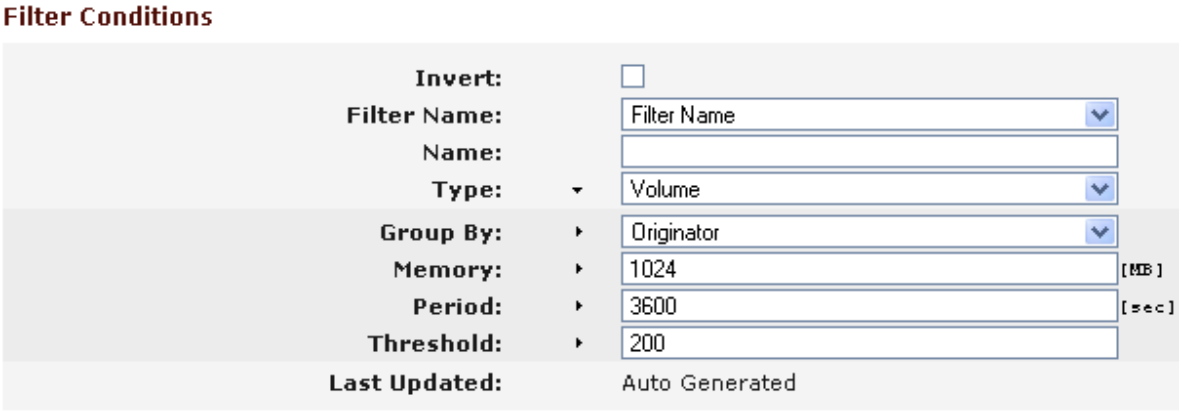


Figure 12: Volume condition MGR configuration

For the volume condition, you can configure the following parameters in the MGR:

- **Group By**—Defines the key fields[1] to group messages by:

Group By	Description	ECI Fields
Nothing	All messages are grouped in a single counting group.	Not applicable

[1] These fields must be included in the ECI evaluation requests that the RTR sends to this FAF.

Group By	Description	ECI Fields
Originator	Messages with the same originator, as specified in the TP-Originating-Address (TP-OA) field of the SM, are grouped and counted.	originatorAddress alphanumericOriginator
Content	Messages with the same content, as specified in the TP-User Data (TP-UD) field of the SM, are grouped and counted. The content is based on raw SMS user data; the user data header is **not** taken into account.	userData
Originator + Content	Messages with both the same originator and content are grouped and counted.	originatorAddress alphanumericOriginator userData

Note: The `NULL` originator is also grouped. Therefore, it is possible to match SRISM with the originator value `"NULL"` using the volume filter. If this is not desired, a filter condition can be added in front of the volume condition to avoid SRISM with `NULL` originator from entering the condition. For example, a expression filter on message type before the volume filter (`messagetype==1`) will only match MO messages.

- **Memory**—The amount of memory in megabytes (MB) to dedicate for the tracking of counting groups in the filter. Valid values range from 256 to 65,536 MB. Refer to *Volume Filtering* for more information.
- **Period**—The tracking period is configurable in seconds, ranging from one minute (60 seconds) up to one day (86,400 seconds). Whenever a message with a new combination of key fields arrives, a new counting group is created, and the tracking period starts for that counting group. At the end of the period, the counting group is reset (discarded from memory).
- **Threshold**—The number of messages in each counting group after which the condition shall apply (return true). Valid values range from 0 to 2,147,483,647 messages.

Possible errors on filter condition activation are:

- Misconfigured parameters—The filter will return an SNMP error and will refuse to activate.
- License check failure—The filter will return an SNMP error and will refuse to activate.

The FAF supports up to **10** instances of the volume condition.

4.6.1 Memory Dimensioning

In certain situations, the volume condition requires a large amount of memory in order to operate accurately.

The memory available to a single volume condition is limited by:

- The amount of free physical memory on the system
- The memory requirements of other processes running on the same system
- The memory requirements of other conditions of the same FAF process, including other volume conditions.

To prevent the volume condition from consuming an undesirably high amount of memory, the volume condition provides a configurable memory limit. Generally speaking, the lower the memory limit is set, the less accurate the volume condition will work. Therefore, it is important to determine and configure a sensible value for this parameter.

Important: Each individual volume condition has its own memory limit parameter.

The formula for determining a value that provides accurate condition evaluation is:

```
Memory [MB] = input_volume * weight (key_fields)
```

The `input_volume` is the number of times the condition is expected to be evaluated in the configured period. If the period is one day, and 1 million messages hit the condition each day, then `input_volume` would be 1 million. The `input_volume` may depend on the applied load distribution (see *Volume Condition Load Distribution*).

The `weight` is a constant value, depending on the key fields setting:

Key Fields	Weight
Nothing	Not applicable [2]
Originator	0.00022
Content	0.00025
Originator + Content	0.0003

If the sum of all memory limits of all volume conditions of one FAF instance exceed the amount of available memory:

- Memory can be traded for accuracy (scale back the **Memory** parameter to fit the available memory), or
- The **Period** and **Threshold** values can be scaled down to reduce the memory requirements, or
- Additional traffic elements can be added to the network to make more memory available.

Example

Using the following input data:

- `input_volume` = 10,000,000 (10 million messages per day)
- `weight` = 0.0003 (Originator + Content)

The memory to be configured is calculated as follows:

```
Memory [MB] = input_volume * weight (key_fields)
            = 10,000,000 * 0.0003
            = 3000 MB
```

Note: The memory value can only be changed when the filter is disabled. Re-enabling the filter again causes the internal data structures to re-dimension according to the newly set parameter for optimal performance and causes all current counting groups to be removed from memory.

When memory is changed from a high value to a low value, the FAF must relocate the current in-use memory from the big memory block to a newly allocated small memory block. This CPU and memory

[2] Because the volume condition with key field "Nothing" requires little memory, it is recommended to set the memory limit to the minimum of 256 MB for these conditions.

load operation can take more than 3 seconds. This could possibly cause the watchdog to kill the FAF. Therefore, it is highly recommended to stop the FAF first, and then change the memory size from a high value to a low value.

4.6.2 Volume Condition Examples

This section provides some configuration examples for the volume filter condition.

Example: Group by Originator

To block traffic exceeding 50 SMS/day from the same originator, the FAF volume condition parameters are set as follows:

- **Type:** Volume
- **Group By:** Originator
- **Memory:** 2200 MB (assuming `input_volume` = 10,000,000)
- **Period:** 86400 seconds (one day)
- **Threshold:** 50 messages

Example: Group by Originator + Content

To block traffic exceeding 30 SMS/12 hours where both the originator and message content are the same, the FAF volume condition parameters are set as follows:

- **Type:** Volume
- **Group By:** Originator + Content
- **Memory:** 1500 MB (assuming `input_volume` = 5,000,000)
- **Period:** 43200 seconds (12 hours)
- **Threshold:** 30 messages

Example: Group by Content

To block traffic exceeding 50 SMS/6 hours where the message content is the same, the FAF volume condition parameters are set as follows:

- **Type:** Volume
- **Group By:** Content
- **Memory:** 625 MB (assuming `input_volume` = 2,500,000)
- **Period:** 21600 seconds (6 hours)
- **Threshold:** 50 messages

Example: Group by Nothing

To block traffic exceeding 1,200,000 SMS/day from a specific country regardless of originator or message content, the FAF volume condition should be triggered by a MTOX rule with:

- **Originator [cond]:** Country + specified country
- **Message Type [cond]:** Bit String + 0 - Short Message
- **EC Application:** External Condition + specified FAF application

- **Failure Action:** Discard with temporary error

The FAF volume condition parameters are set as follows:

- **Type:** Volume
- **Group By:** Nothing
- **Memory:** 256 MB
- **Period:** 86400 seconds (one day)
- **Threshold:** 1200000 messages

4.6.3 Volume Condition Load Distribution

The volume condition counts SMs with the same key fields. For accurate operation, it is important that SMs with the same key fields get processed by the same FAF instance. This can be achieved by controlling the load distribution as applied by the RTR.

Key Fields	Load Distribution
Nothing	Any [3]
Originator	Key-based by originator address
Content	None [4]
Originator + Content	Key-based by originator address

Each FAF instance tracks its own set of counting groups. Whenever load distribution causes counting groups for the same key fields to be created on multiple FAF instances, the threshold applying to the whole network needs to be divided by the number of FAF instances. Also, in such situations, the accuracy of the volume condition degrades. Degradation gets stronger, the lower the threshold.

Important: In case FAF instances are chained to achieve the configuration as mentioned in *Volume Condition Examples*, threshold levels of the condition with key fields "nothing" may need revision. Note that statistically, when given high traffic volumes the traffic will be most likely equally spread, given high thresholds (for example, greater than 10,000) and an accepted accuracy margin, a volume condition with key fields "content" can produce acceptable results also when handled across multiple FAF instances. Please take into account that traffic characteristics may vary per network and per country.

4.6.4 Volume Condition Required Message Fields

The FAF receives evaluation requests via the ECI interface, through which it connects to a RTR.

For the volume condition, the external condition (EC) application must be configured to send the message fields specified in the volume condition's **Group By** parameter to the FAF. The following message fields apply (refer to the table in *Volume Filtering*):

- Originator Address

[3] When the SMs are distributed over *N* FAF instances, the configured volume condition threshold should calculated by dividing the threshold to be applied to the total amount of SMs by *N*.

[4] Currently, the RTR is not able to distribute the load in such a way that messages with equal content end up on the same FAF instance. Therefore, a volume condition with key fields "content" can only work accurately if configured such that (also) a single instance of the FAF handles all SMs.

- Alphanumeric Originator
- User Data

4.6.5 Volume Condition Traps

The volume condition may issue the following SNMP traps:

- `volumeStartDetected`—The volume condition starts to apply action on the matched counting group.
- `volumeEndDetected`—The volume condition stops to apply action on the matched counting group.
- `volumeMemoryInUseAlert`—Alerts the operator that the memory in use for a volume condition reaches the memory limit as configured in the **Memory** field of the volume filter condition.
- `volumeMemoryInUseAlertClean`—Informs the operator that the memory in use for a volume condition has dropped 50 MB below the configured memory limit.

Refer to the ZephyrTel Mobile Messaging SNMP Trap Reference Guide for more information about these traps.

4.6.6 Volume Counters Reset for Blacklist Subscriber

The FAF will reset the volume filter counter in the following scenario:

1. **Daily Reset**: If this parameter is set as an absolute time, the per volume counter of a subscriber is going to be reset at specified time, which is configured in **Reset at**, in the MGR. If the parameter is set to `none`, the volume counter will not be reset on a specified time.
2. **Reset at specific time interval**: FAF will reset the volume counter at the specified time. The specific time interval is provided by the user through the MGR.

   This functionality allows the message counters of the originator subscribers to be reset at the specified time (in `HHMMSS`), every day.

Figure 13: Configuration of the specific time through MGR as Daily Reset Time.

3. **Reset Volume Counter after successful Response from SPF**: FAF will clear the volume counters after getting the successful response for a blacklist request from SPF. When the semi-static parameter *fafenableclearvolumecounterforblacklist* is set to "true", FAF will reset the volume counter for the subscriber. If this parameter is "false" the existing behavior will be followed, i.e. volume counters will not be reset.

4.7 Bulk Filtering

The FAF's bulk condition detects messages with matching fields that FAF receives during a relatively short timespan; for example, several seconds or several minutes. The field that the FAF checks for matching can be a message field such as originator, recipient, SMSC, and so on.

You can use the bulk condition to detect that a single originator is sending messages in bulk, even if the originator is not sending messages at a regular rate, and/or is sending messages more slowly than what the flooding condition will detect.

You can also use one or more bulk conditions to reduce the load on the duplicates condition. The FAF has a limited buffer for tracking duplicates clusters, so reducing the number of messages that it must evaluate improves its performance.

Note: For best results, a flooding condition should be provisioned with a higher priority than the bulk condition.

Because the message rate may vary, FAF calculates the average timespan between each message and the message that came before it. When the average timespan crosses a configured threshold, FAF considers this to be bulk messaging, and returns "true" for the condition.

FAF uses internal records to track the messages that match the bulk condition. To prevent spurious marking of bulk messages, FAF expires its internal records after a configured time period.

The FAF can track 2^{19} (524288) bulk records. This equals to 145 message per second for 3600 seconds. Either reducing the validation period or reducing message per second can avoid overflow.

You can use the `tp_walk` command-line tool to view up to 500 of the records that the FAF generates during bulk detection in the `fafBulkCountTable` SNMP table. Although the FAF can trace up to 524,288 records at once, it only stores 500 of the records in the SNMP table. This preserves resources and prevents significant delays when `tp_walk` is used.

The FAF supports up to **10** instances of the bulk condition.

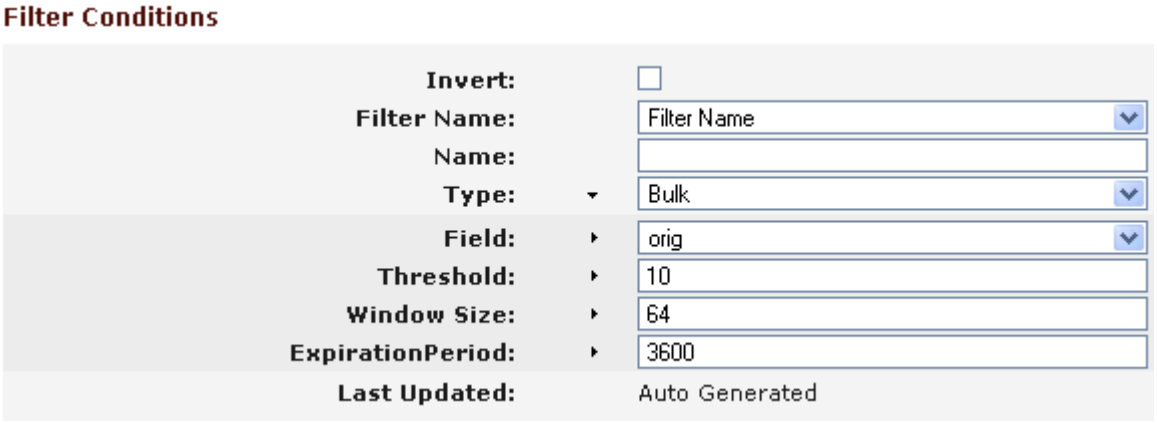


Figure 14: Bulk condition MGR configuration

4.7.1 Bulk Condition Calculation

The bulk condition uses the autoregressive moving average model (ARMA) to calculate the average time between messages. This section explains the calculation.

This figure illustrates a timeline during which FAF is receiving messages with matching fields (for example, messages from the same originator).

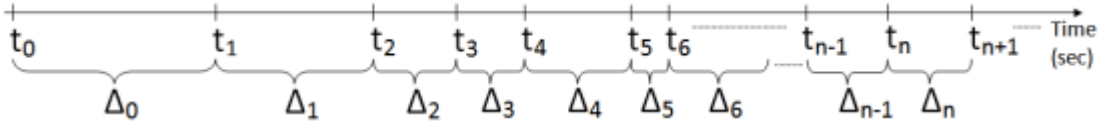


Figure 15: Bulk condition calculation

To determine the average timespan between each message, FAF calculates:

Variable	Represents...
t_n	The moment that FAF receives a message with matching fields, expressed in seconds. As soon as the filter containing the bulk condition is activated, the timeline starts at 0. For example, if a message arrives 10 seconds after filter activation, t_0 is equal to 10. If a second message with matching information then arrives 25 seconds after filter activation, t_1 is equal to 25.
$\Delta_n = t_{n+1} - t_n$	The timespan between two messages with matching fields. This value changes dynamically because it is recalculated every time a new message with matching fields is received. For example, if t_0 is 10 and t_1 is 25, then this value is: $\Delta_0 = t_1 - t_0 = 25 - 10 = 15$
$A_0 = 2 * \text{threshold}$	The initial average value when the first message is received. This value is always two times the configured threshold.

The formula to calculate the average at a specific moment when a message with matching information is received is:

$A_n = A_{n-1} * C + \Delta_{n-1} * (1 - C)$

Which can also be written as:

new_value = old_value * C + timespan * (1 - C)

Where:

Variable	Represents...
A_n	The average time at n
C	$C = \exp\left(-\frac{previous\ timespan}{window}\right) = e^{\left(\frac{-\Delta_{n-1}}{window}\right)} = [0 \dots 1]$

Variable	Represents...
	The auto-regressive moving average model, where *e* is the mathematical constant (approximately 2.71)

A message is marked as bulk when:

A_n < threshold

C is recalculated every time a new message with matching information is received. The value of C is heavily influenced by the timespan between two messages with matching information.

If C is close to 1, FAF is less likely to mark messages as bulk, or it will take some time until messages are marked as bulk. As C decreases in value, the chance that messages are marked as bulk increases.

4.7.1.1 Bulk Condition Calculation Example

This section provides an example of the bulk condition calculation. This example uses the default bulk condition values:

- Field is originator
- Threshold is 10
- Window size is 64
- Expiration is 3600 seconds

Assume messages from a single originator are arriving at the following times:

$t_0 = 0$

$t_1 = 20$

$t_2 = 30$

$t_3 = 40$

$t_4 = 45$

$t_5 = 50$

$t_6 = 60$

The initial average value is:

$A_0 = 2 * \text{threshold} = 2 * 10 = 20$

The timespan between each message is:

$\Delta_0 = t_1 - t_0 = 20 - 0 = 20$

$\Delta_1 = t_2 - t_1 = 30 - 20 = 10$

$\Delta_2 = t_3 - t_2 = 40 - 30 = 10$

$\Delta_3 = t_4 - t_3 = 45 - 40 = 5$

$\Delta_4 = t_5 - t_4 = 50 - 45 = 5$

$\Delta_5 = t_6 - t_5 = 60 - 50 = 10$

The average value calculations are:

Message	Formula	Calculation	Marked as bulk?
1	$A_1 = A_0 * C + {}_0 * (1 - C)$, where $C = \exp(-20/64) = e^{(-20/64)} \approx 0.73$	$A_1 = A_0 * C + \Delta_0 * (1 - C) = 20 * 0.73 + 20 * (1 - 0.73) = 14.6 + 5.4 = 20$	No, because A_1 is greater than the threshold (20 > 10)
2	$A_2 = A_1 * C + \Delta_1 * (1 - C)$, where $C = \exp(-10/64) = e^{(-10/64)} \approx 0.86$	$A_2 = A_1 * C + \Delta_1 * (1 - C) = 20 * 0.86 + 10 * (1 - 0.86) = 17.2 + 1.4 = 18.6$	No, because A_2 is greater than the threshold (18.6 > 10)
3	$A_3 = A_2 * C + \Delta_2 * (1 - C)$, where $C = \exp(-10/64) = e^{(-10/64)} \approx 0.86$	$A_3 = A_2 * C + \Delta_2 * (1 - C) = 18.6 * 0.86 + 10 * (1 - 0.86) = 15.99 + 1.4 = 17.3$	No, because A_3 is greater than the threshold (17.3 > 10)
4	$A_4 = A_3 * C + \Delta_3 * (1 - C)$, where $C = \exp(-5/64) = e^{(-5/64)} \approx 0.93$	$A_4 = A_3 * C + \Delta_3 * (1 - C) = 17.3 * 0.93 + 5 * (1 - 0.93) = 16.08 + 0.35 = 16.53$	No, because A_4 is greater than the threshold (16.53 > 10)
5	$A_5 = A_4 * C + \Delta_4 * (1 - C)$, where $C = \exp(-5/64) = e^{(-5/64)} \approx 0.93$	$A_5 = A_4 * C + \Delta_4 * (1 - C) = 16.53 * 0.93 + 5 * (1 - 0.93) = 15.37 + 0.35 = 15.72$	No, because A_5 is greater than the threshold (15.72 > 10)
6	$A_6 = A_5 * C + \Delta_5 * (1 - C)$, where $C = \exp(-10/64) = e^{(-10/64)} \approx 0.86$	$A_6 = A_5 * C + \Delta_5 * (1 - C) = 15.72 * 0.86 + 1 * (1 - 0.86) = 13.51 + 0.02 = 13.53$	No, because A_6 is greater than the threshold (13.53 > 10)

As of t_6, no message has been marked as bulk, so the recommendation is to increase the threshold. Assuming the threshold is increased to 50, the average value calculations are:

Message	Formula	Calculation	Marked as bulk?
1	$A_1 = A_0 * C + \Delta_0 * (1 - C)$, where $C = \exp(-20/64) = e^{(-20/64)} \approx 0.73$	$A_1 = A_0 * C + \Delta_0 * (1 - C) = 100 * 0.73 + 20 * (1 - 0.73) = 73 + 5.4 = 78.4$	No, because A_1 is greater than the threshold (78.4 > 50)
2	$A_2 = A_1 * C + \Delta_1 * (1 - C)$, where $C = \exp(-10/64) = e^{(-10/64)} \approx 0.86$	$A_2 = A_1 * C + \Delta_1 * (1 - C) = 78.4 * 0.86 + 10 * (1 - 0.86) = 67.4 + 1.4 = 68.8$	No, because A_2 is greater than the threshold (68.8 > 50)
3	$A_3 = A_2 * C + \Delta_2 * (1 - C)$, where $C = \exp(-10/64) = e^{(-10/64)} \approx 0.86$	$A_3 = A_2 * C + \Delta_2 * (1 - C) = 68.8 * 0.86 + 10 * (1 - 0.86) = 59.1 + 1.4 = 60.5$	No, because A_3 is greater than the threshold (60.5 > 50)
4	$A_4 = A_3 * C + \Delta_3 * (1 - C)$, where $C = \exp(-5/64) = e^{(-5/64)} \approx 0.93$	$A_4 = A_3 * C + \Delta_3 * (1 - C) = 60.5 * 0.93 + 5 * (1 - 0.93) = 56.2 + 0.35 = 56.6$	No, because A_4 is greater than the threshold (56.6 > 50)
5	$A_5 = A_4 * C + \Delta_4 * (1 - C)$, where $C = \exp(-5/64) = e^{(-5/64)} \approx 0.93$	$A_5 = A_4 * C + \Delta_4 * (1 - C) = 56.6 * 0.93 + 5 * (1 - 0.93) = 52.6 + 0.35 = 52.9$	No, because A_5 is greater than the threshold (52.9 > 50)
6	$A_6 = A_5 * C + \Delta_5 * (1 - C)$, where $C = \exp(-10/64) = e^{(-10/64)} \approx 0.86$	$A_6 = A_5 * C + \Delta_5 * (1 - C) = 52.9 * 0.86 + 10 * (1 - 0.86) = 45.49 + 1.4 = 46.8$	Yes, because A_6 is smaller than the threshold (46.8 < 50)

4.7.2 Bulk Condition Use Case

This section describes a use case in which the bulk condition is used to reduce the load on the duplicates condition.

When using the bulk condition to reduce the load on the duplicates condition, you assume that:

- An originator who is not sending a large number of messages in a short period of time is not sending duplicate messages, so the FAF does not need to track those messages as part of a duplicates cluster
- A recipient who is not receiving a large number of messages in a short period of time is not receiving duplicate messages, so the FAF also does not need to track those messages as part of a duplicates cluster

For example, this use case assumes that:

- An originator who is not sending more than one message per minute (on average) is not sending duplicate messages
- A recipient who is not receiving more than one message every 30 seconds (on average) is not receiving duplicate messages

This figure illustrates how two filters can be used to check the originator's rate of sending, check the recipient's rate of receiving, and check for duplicates. The first filter has a higher priority.

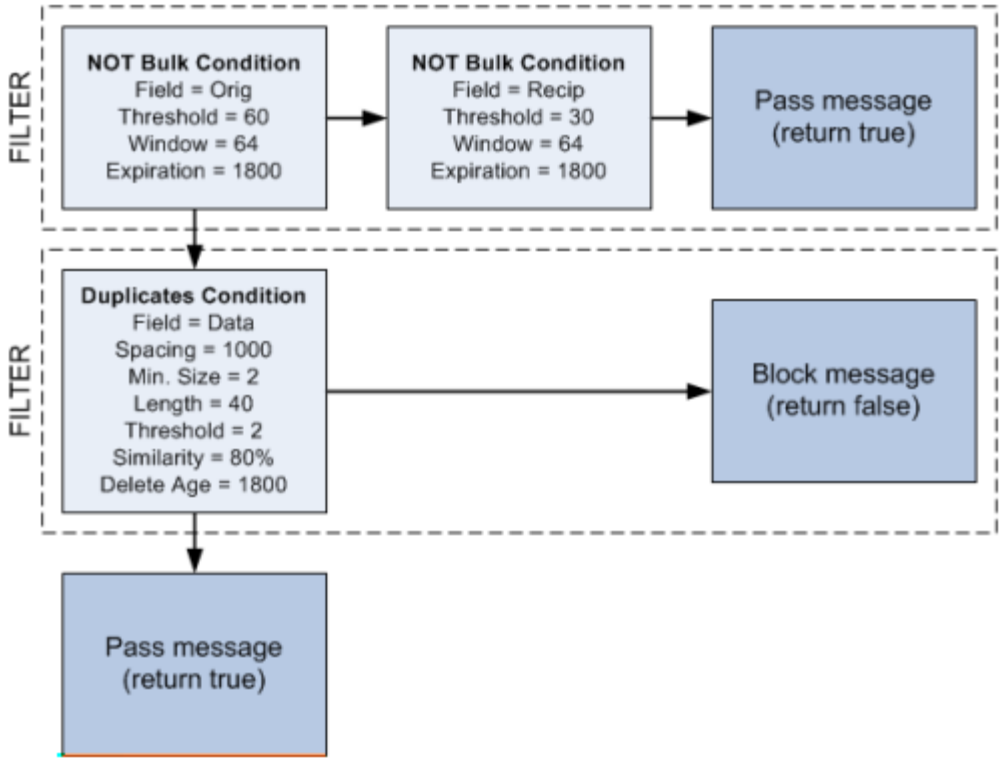


Figure 16: Sample filters with bulk and duplicates conditions

The filters operate as follows:

Filter	Description
First filter	The filter with the highest priority contains two bulk conditions: **1.** The first condition checks if the message originator is not sending messages less than 60 seconds apart (on average) **2.** The second condition checks if the message recipient is not receiving messages less than 30 seconds apart (on average) If: • Either condition is true, pass the message that is being evaluated (FAF returns "true" to the RTR) • Both conditions are true, pass the message that is being evaluated (FAF returns "true" to the RTR) • Neither condition is true, proceed to the filter with the next-lowest priority
Second filter	The filter with the next-lowest priority contains one duplicates condition, which checks for a duplicates cluster that contains messages with similar content. If the message: • Is a duplicate, block the message that is being evaluated (FAF returns "false" to the RTR) • Is not a duplicate, proceed to the next filter with the next-lowest priority

Combining Bulk and Duplicates Filtering with Content Filtering

You can combine the bulk and duplicates filtering illustrated above with a content condition that checks messages for words or phrases that should be blocked. This figure illustrates how a content condition can be added to the bulk and duplicates conditions.

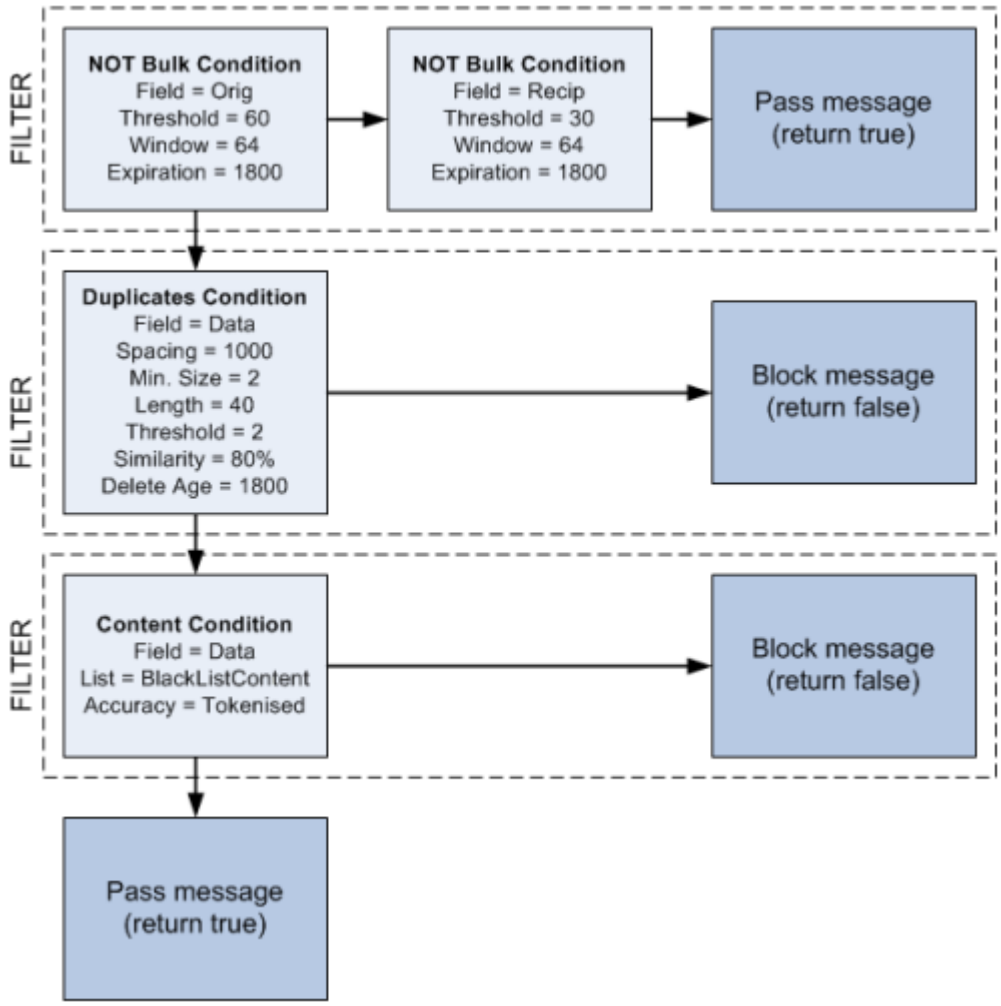


Figure 17: Sample filters with bulk, duplicates, and content conditions

The filters operate as follows:

Filter	Description
First filter	The filter with the highest priority contains two bulk conditions: **1.** The first condition checks if the message originator is not sending messages less than 60 seconds apart (on average) **2.** The second condition checks if the message recipient is not receiving messages less than 30 seconds apart (on average) If: • Either condition is true, pass the message that is being evaluated (FAF returns "true" to the RTR) • Both conditions are true, pass the message that is being evaluated (FAF returns "true" to the RTR)

Filter	Description
	• Neither condition is true, proceed to the filter with the next-lowest priority
Second filter	The filter with the next-lowest priority contains one duplicates condition, which checks for a duplicates cluster that contains messages with similar content. If the message: • Is a duplicate, block the message that is being evaluated (FAF returns "false" to the RTR) • Is not a duplicate, proceed to the next filter with the next-lowest priority
Third filter	The filter with the lowest priority contains one content condition, which checks the message content (user data) for words or phrases that the operator wants to block. If the message: • Contains blacklisted content, block the message that is being evaluated (FAF returns "false" to the RTR) • Does not contain blacklisted content, pass the message that is being evaluated (FAF returns "true" to the RTR)

4.7.3 Bulk Condition Required Message Fields

For the bulk condition, the external condition (EC) application must be configured to send the message field specified in the bulk condition's **Field** parameter to the FAF.

4.7.4 Bulk Condition Traps

The bulk condition may issue the following SNMP traps:

- `bulkStartDetected`—A new bulk record has been detected.
- `bulkEndDetected`—No more bulk messages have been detected for this value of the filter variable.

Refer to the ZephyrTel Mobile Messaging SNMP Trap Reference Guide for more information on the traps.

4.8 Enhanced Messaging (EMS) Filtering

The FAF's enhanced messaging (EMS) condition detects messages that contain specific Information Element IDs (IEIs) in the user data header (UDH) or specific protocol id values. If the UDH of a message contains the selected IEI(s) or if the message contains a protocol id that matches with any of the configured **Protocol Id Values**, the filter condition will return "true". If the **Protocol Id Values** parameter is left blank, then the filter condition will match any protocol id value.

The EMS condition uses a logical OR operation; therefore, if you select multiple IEIs and also configure certain protocol id values for one EMS condition and the message contains any of the selected IEIs OR any of the configured protocol ids, the condition will return "true".

In the case of the application port addressing scheme IEIs, FAF can also verify that the source and destination ports in these IEIs match provisioned ports. This functionality can be used, for example, to detect and block WAP push messages, which use 16-bit source port 9200 (decimal) and destination port 2948 (decimal).

Refer to chapter 9.2.3.24 of the 3GPP 23.040 specification for a description of EMS IEIs.

The FAF supports up to **100** instances of the EMS condition.

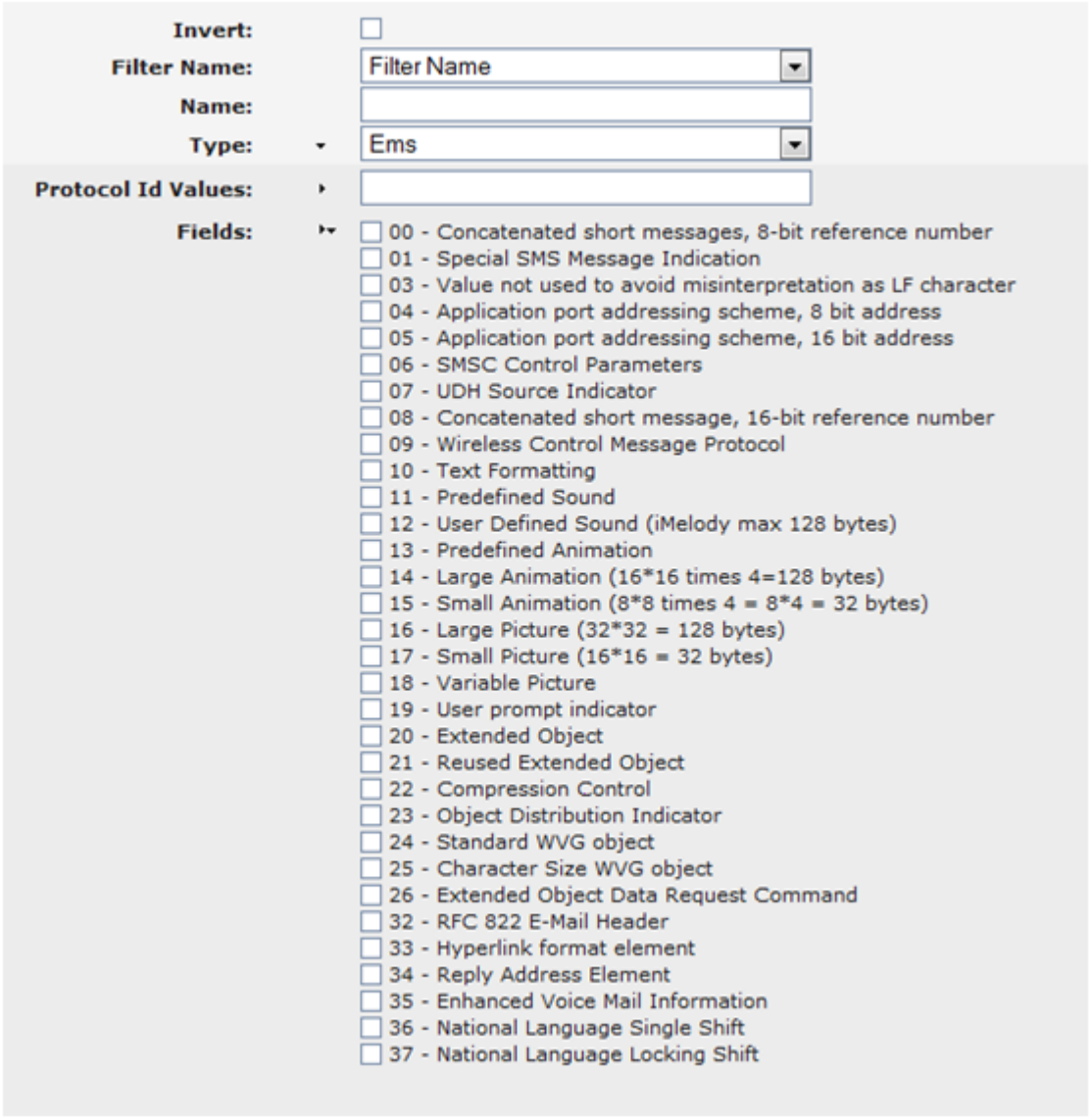


Figure 18: EMS condition MGR configuration

4.8.1 EMS Condition Example

You can use the EMS condition to check for the presence of a Wireless Application Protocol (WAP) push message:

- To check for a connectionless WAP browser proxy server, select the **05 - Application port addressing scheme, 16 bit address** IEI and enter 9200 in the **16 Bit Source Port** box. If the source port in the UDH of a message is 9200, the FAF will return "true" for the condition.
- To check for the WAP push connectionless session service, select the **05 - Application port addressing scheme, 16 bit address** IEI and enter 2948 in the **16 Bit Destination Port** box. If the destination port in the UDH of a message is 2948, the FAF will return "true" for the condition.

4.8.2 EMS Condition Required Message Fields

For the EMS condition, the external condition (EC) application must be configured to send the user data header message field and, if the **Protocol Id Values** parameter is not blank, the “protocolId” message field as well, to the FAF.

4.8.3 EMS Condition Traps

The EMS condition does not generate SNMP traps.

4.9 Expression Filtering

The FAF's expression condition can test the value of a field and/or assign a value to a field. You can use the expression condition to assign values to external attributes (`eciattribute` fields).

FAF's test expression condition can be used to:

- apply filter on messages of certain message type
- have conditional evaluation based on set ECI attributes
- set a limit on the number of segments of a concatenated SMS.

Please refer to section *Add an Expression Condition* for available syntax and variables for the assignment and test expression.

You can nest expressions by using parentheses. For example:

```
(messagetype > 1 && messagetype < 4 && totalsegments > 4)
```

The FAF supports up to 100 instances of the expression condition.

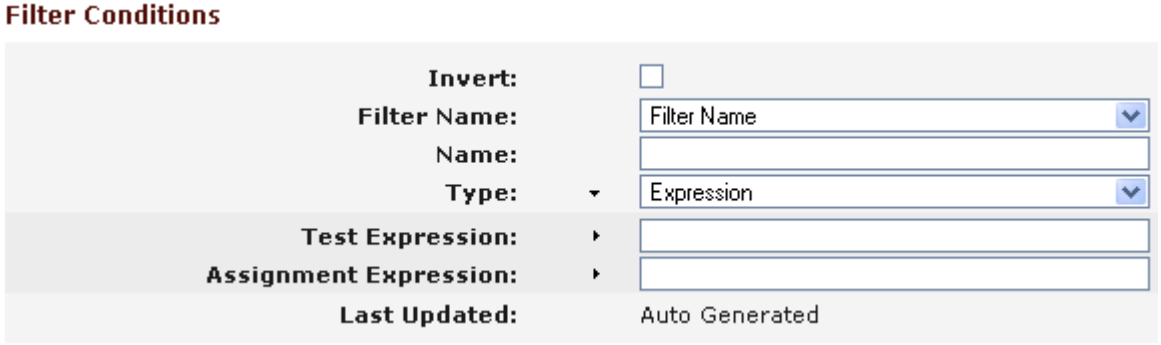


Figure 19: Expression condition MGR configuration

4.9.1 Expression Condition Traps

The expression condition does not generate SNMP traps.

4.10 Delta Filtering

The FAF's delta condition allows blocking of short messages (SMs) with certain characteristics, which are classified as "sim-box".

The delta condition detects and measures the difference of a single mobile MSISDN as originator and recipient. If the Mobile subscriber sends lot of messages and receive very few or none (For example received is <1% of message sent) then it is more likely that the Mobile subscriber is a sim-box.

For this filter, FAF will record the total number of messages and delta count for Mobile subscribers. If the volume of the SMS is beyond a certain level (e.g. 500 SMS) and the ratio between SMS sent and SMS received is below a certain percentage (e.g. 5%), the FAF Filter will match. There are two phases to measure the originator and recipient:

- Phase 1: Originating Leg
  - On matching a MOX/AOX rule, the RTR load-balances ECI messages among FAF based on the Originator number as key.
  - For each ECI message, FAF will determine the entry based on the Originator.
    - Originator and Recipient Address must be MSISDN (not Short Code, Alphanumeric)

      Note: The EC Rule for FAF and the FAF Chain shall be configured to prevent short code and alphanumeric addresses from being routed to the FAF and analyzed by the Delta Filter.

    - Address must be in normalized format (EC Application supports specifying Originator and Recipient format)
  - If the entry exists, the total messages sent count is incremented by 1. Otherwise a new entry is created, and total message sent count is set to 1.
  - Verify if the Delta Filter condition is matched for the MSISDN. The condition is reached if both conditions are true:
    - The volume of messages is more than Message Limit configured
    - The Receive Sent percentage is less than or equal to the configured percentage.

- Phase 2: Terminating Leg
  - On matching MTOX/ATOX rule, RTR load-balance ECI messages among FAF based on the Recipient number as key.
  - If Delta filter is active, and message type "mtShortMessage", "atShortMessage", FAF will request for ECI Indication. FAF will not perform any processing for Delta Filter at this moment.
  - RTR will attempt delivery of the message. On success or failure, it will send ECI indication to FAF.
  - When ECI indication is received and status is "success", then for Delta Filters, FAF will determine the entry based on the Recipient.

  - Address must be in normalized format (EC Application supports specifying Originator and Recipient format)

- If the entry exists, then total message received count is incremented by 1. Else new entry is created, and total message received count is set to 1.
- No other action is performed as message is already delivered.

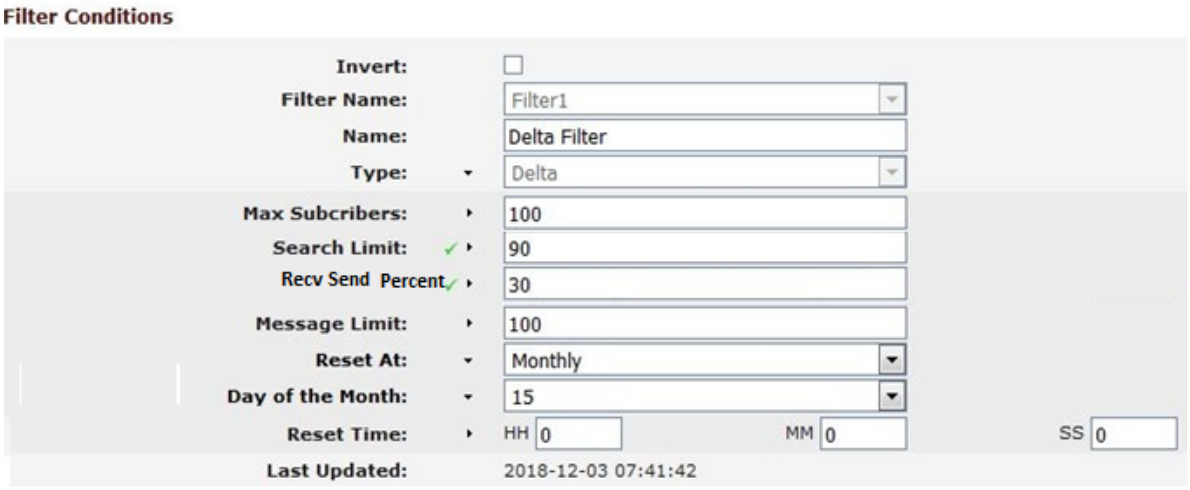


Figure 20: Sample delta filter condition

For the delta condition, you can configure the following parameters in the MGR:

- **Max Subscribers**: The maximum number of subscriber numbers the delta filter can contain. Entries are replaced on a least recently used (LRU) basis. This variable can only be changed when the admin state is `inactive`. The valid range is 1-64800000 and the default value is 864000.
- **Search Limit**: The number of records to search for a free slot in the maintained subscriber list by the Delta filter where the new record can be inserted. The starting point of the search in the list will be the index calculated on the basis of hash value of MSISDN contained in the message. Increasing this value will increase the accuracy of search in the subscriber list but will also increase CPU time for the delta filter. This variable can only be changed when the admin state is `inactive`. The valid range is 1-999 and the default value is 16.
- **Recv Send Percent**: Ratio percentage between SMS received and SMS sent by a subscriber. If the ratio percentage is equal to or less than the configured value of this field and the total number of messages for that subscriber is equal to or greater than the value of the configured **Message Limit**, the filter matches. This variable can be changed when the admin state is `active`. The valid range is 0-100 and the default value is 5.
- **Message Limit**: The limit of the number of messages per subscriber. If the total messages number (send + receive) is equal to or above this value and the ratio percentage between SMS received and SMS sent is lower than or equal to the configured value of the field **Recv Send Percent**, the delta filter returns true. This variable can be changed when the admin state is `active`. The valid range is 0-2147483647 and the default value is 1000.
- **Reset At**: The delta filter can be reset at three possible frequency periods:
  - **Daily**: This indicates that counters will be reset daily.
  - **Weekly**: This indicates that counters will be reset once a week based on configured day of the week.

- **Monthly**: This indicates that counters will be reset once a month based on configured day.

Note:

1. The **Reset At** period can be updated even if the filter's admin state is `active`.
2. If the configured day of the month is not present in a month, the reset will not happen in that month. E.g. if the configured day is 31st, then the reset will not happen in the month of February, April, etc.
3. If the reset has occurred once for a day, if the configured reset-period is **Daily**, the updated reset-time will be applicable from the next day onwards. If the change is required to be in-effect immediately, then the re-activation of the filter condition is required.

4.10.1 Memory Consideration

The memory that a delta condition uses is linear with respect to the configured "maximum subscribers". Therefore, calculate the memory usage with the following formula:

```
Total memory = "size of a single subscriber record"  x  "maximum subscribers"
```

The "`size of a single subscriber record`" includes the following parts:

- MSISDN (41 bytes) for 20 digits
- Length of the MSISDN (1 byte)
- Timestamp of the last message (4 bytes)
- Timestamp of the first message (4 bytes)
- Total number of sent messages (4 bytes)
- Total number of received messages (4 bytes)

For example, the "size of a single subscriber record" is:

```
41 + 1 + 4 + 4 + 4 + 4 = 58
```

It is rounded up to 60 bytes for address alignment.

Therefore, the total memory of one million subscriber is: 60 BYTE * 1,000,000 is around 60MB.

Warning: FAF also has other memory overheads for the delta condition. Therefore, the above calculate provides an indication of the range of memory consumption. The real memory size will be bigger than the above calculated number.

4.10.2 RTR EC Application Configuration

The delta condition requires the "dynamic" RTR EC application distribution key.

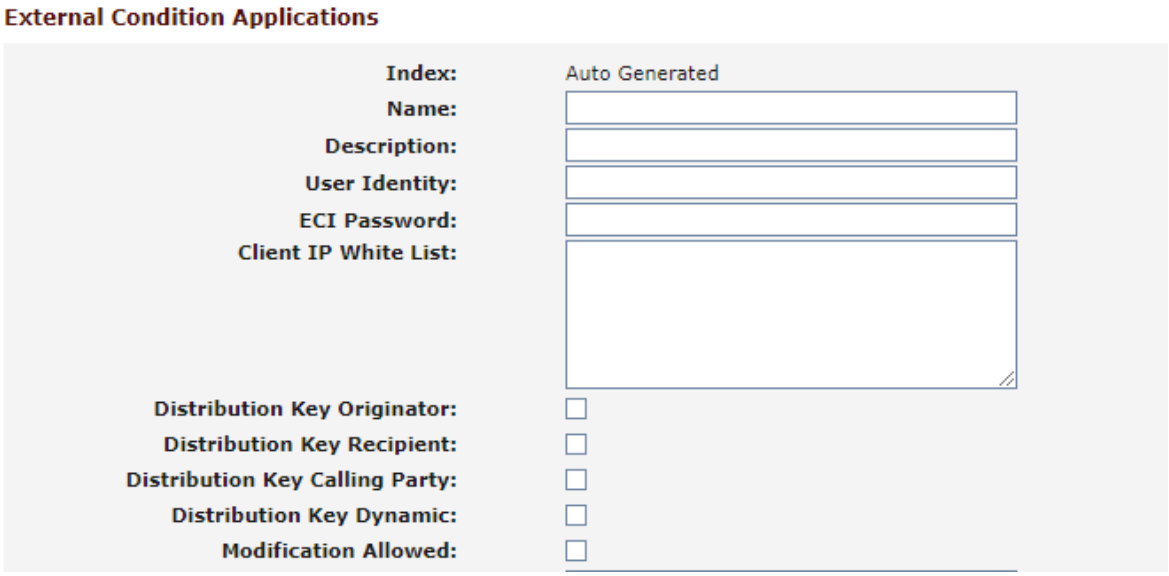


Figure 21: RTR EC Application configuration

If the distribution key is not dynamic, the outcome of the delta filter condition is unspecified.

4.10.3 Incompatibility With Other FAF Conditions

The delta condition working approach is not compatible with the other FAF conditions. This is because it not only requires different EC application distribution key, but also the fact that a single message will be received by the FAF twice for both originator and recipient counting.

There are two kinds of FAF filter conditions:

1. Those decision making on an individual message like content, EMS and expression filters.
2. Those decision making on historical information of the messages satisfying the same conditions, like bulk, flooding, duplicate and volume filters.

For FAF conditions in category (1), the blocking logic is not affected, but we are wasting FAF resource by doing the match twice. For FAF conditions in category (2), the blocking logic is broken by recording the message twice.

Therefore, it is highly recommended to configure FAF instances to perform the delta condition alone without the other FAF filter conditions.

4.11 Spread Filtering

The FAF's spread condition allows blocking of short messages (SMs) with certain characteristics, which are classified as "sim-box".

The spread condition detects and measures the spread of messages from a single Mobile Originator.

On matching a MOX/AOX rule, the RTR load-balances ECI messages among the FAFs based on the Originator number as key.

- The Originator Address must be a MSISDN (not Short Code or Alphanumeric).

  Note: The EC Rule for FAF and the FAF Chain shall be configured to prevent short code and alphanumeric addresses from being routed to the FAF and analyzed by the Spread Filter.

- The address must be in normalized format (EC Application supports specifying Originator and Recipient format).

Spread is calculated as the percentage of different recipients by total number of messages sent with defined period. Every message to the same recipient is 0% spread, every message to a different recipient is 100% spread. If any Originator sends a lot of messages with high spread, then it is more likely that the Originator is a sim-box. Therefore, the spread percent is defined as:

```
(number of different recipients / total number of message) *100
```

Rather than taking spread as percentage, FAF records the total number of messages and the recipient spread for the same originator. The definition of the two parameters are the following:

1. The "message limit" that is the total message count;
2. The "recipient limit" that is the number of different recipients.

When both of the following two conditions are satisfied:

1. The total messages exceed a certain threshold (e.g. 300 per day),
2. The number of different recipients is above a certain threshold (e.g. 100),

then the FAF spread condition is matched.

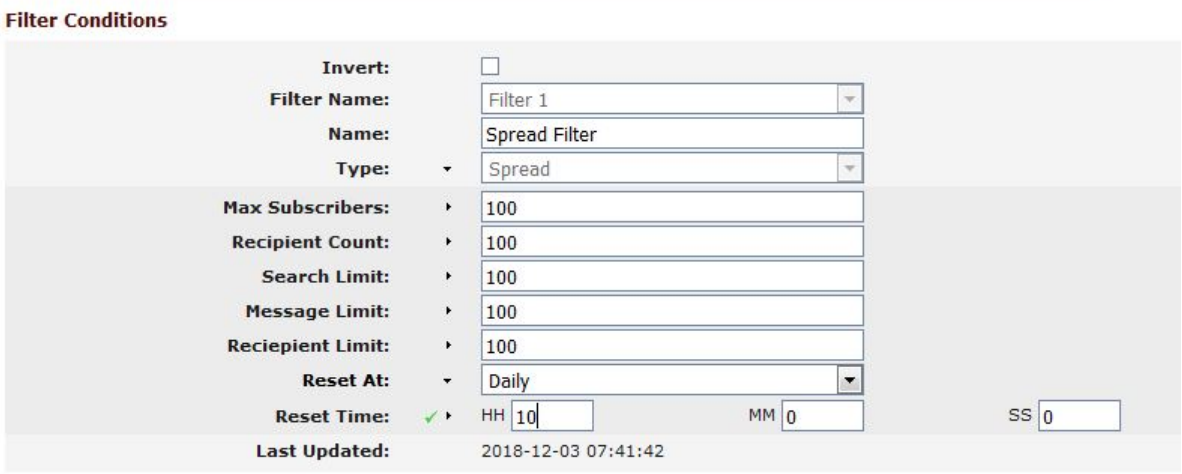


Figure 22: Sample spread filter condition

For the spread condition, you can configure the following parameters in the MGR:

- **Max Subscribers**: The maximum number of originator numbers the spread filter can contain. Entries are replaced on least recently used (LRU) basis. This variable can only be changed when the admin state is `inactive`. The valid range is 1-64800000 and the default value is 864000.
- **Recipient Count**: The size of the recipient bitmap. The recipients are hashed via a hash function to an entry in the recipient bitmap. It is recommended to set this to at least thrice the size of the **Recipient Limit**. This variable can only be changed when the admin state is `inactive`. The size of the bitmap is internally rounded up to a multiple of 32. The valid range is 0-99999 and the default value is 1000.

- **Search Limit**: The number of records to search for a free slot in the maintained subscriber list by Spread filter where the new record can be inserted. The starting point of the search in the list will be the index calculated on the basis of hash value of MSISDN contained in the message. Increasing this value will increase the accuracy of search in the subscriber list but will also increase the CPU time for the spread filter. This variable can only be changed when the admin state is `inactive`. The valid range is 1-999 and the default value is 16.
- **Message Limit**: The limit of the number of messages per subscriber. Above the limit the number of different entries in the recipient bitmap is counted. If the message count and the number of different recipients in the bitmap are above the values of the MsgLimit and RecipLimit, respectively, the spread filter returns true. Setting the fafSpreadMsgLimit below the fafSpreadRecipLimit is not recommended. This variable can be changed when the admin state is `active`. The valid range is 0-2147483647 and the default value is 1000.
- **Recipient Limit**: The limit of different entries in the recipient bitmap that is an approximation of the number of different recipients, the originator has sent their messages to. If the message count and the number of different recipients in the bitmap are above the values of the MsgLimit and RecipLimit, respectively, the spread filter returns true. It is recommended to set the recip limit to less than a third of the recipient count for accurate statistics. This variable can be changed when the admin state is active. The valid range is 1-9999 and the default value is 300.
- **Reset At**: The spread filter can be reset at three possible frequency periods:
  - **Daily**: This indicates that counters will be reset daily.
  - **Weekly**: This indicates that counters will be reset once a week based on configured day of the week.
  - **Monthly**: This indicates that counters will be reset once a month based on configured day.

Note:

1. The **Reset At** period can be updated even if the filter's admin state is active.
2. If the configured day of the month is not present in a month, the reset will not happen in that month. E.g. if the configured day is 31st, then the reset will not happen in the month of February, April, etc.
3. If the reset has occurred once for a day, if the configured reset-period is **Daily**, the updated reset-time will be applicable from the next day onwards. If the change is required to be in-effect immediately, then the re-activation of the filter condition is required.

4.11.1 Memory Consideration

The memory that a spread condition uses is linear with respect to the configured "maximum subscribers". Therefore, calculate the memory usage with the following formula:

```
Total memory = "size of a single subscriber record"  x  "maximum subscribers"
```

The "`size of a single subscriber record`" includes the following parts:

- MSISDN (41 bytes) for 20 digits
- Length of the MSISDN (1 byte)
- Timestamp of the last message (4 bytes)
- Timestamp of the first message (4 bytes)
- Total number of messages (4 bytes)
- Total number of unique recipients (4 bytes)
- Bitmap Array of (Recipient Count / 8) bits // size will be rounded to multiple of 32 bits

For example, for Recipient Count of 1000 (round to 1024), the "size of a single subscriber record" is:

```
41 + 1 + 4 + 4 + 4 + 4 + 1024/8 = 182
```

then round to 184 bytes for address alignment.

Therefore, the total memory of one million subscriber is: 184 BYTE * 1,000,000 is around 184MB.

Warning: FAF also has other memory overheads for the spread condition. Therefore, the above calculate provides an indication of the range of memory consumption. The real memory size will be bigger than the above calculated number. The maximum memory consumed by the spread filter is restricted to 4GB.

4.11.2 Configuration Recommendation

The different recipient count uses bloom filter algorithm. The bloom filter algorithm has a chance to map different recipients to the same key. Therefore, it is possible that the spread condition is not triggered after receiving SMS from the same originator to **Recipient Limit** number of different recipients.

In order to increase the accuracy of the filtering, it is recommended to use Recipient Count that is at least 3 times the value of **Recipient Limit**. The following is one sample configuration:

- `fafSpreadMaxSubscribers`: 864000
- `fafSpreadRecipientCount`: 1000 (slightly more than 3 times the size of recipient limit)
- `fafSpreadSearchLimit`: 16
- `fafSpreadMsgLimit`: 1000
- `fafSpreadRecipLimit`: 300

The above sample configuration starts to mark the originator as sim-box when there are more than 1000 messages from the originator to more than 300 different recipients.

4.12 External Condition Messages

If a flooding or bulk condition is detected from an originating MSISDN to the recipients, it is possible to configure the Router to generate an external condition message (ECM) to the originator containing, for example, “Your handset is suspected to be infected with virus…”.

To generate the ECM, make sure that the `failuremessagekey` is set in the Expression Condition to a value, which is matched on the Router against the provisioned template message.

Refer to *ECM and ABL Configuration Example* for an ECM configuration example.

Chapter 5

OAM Interface (SNMP)

Topics:

5.1 Introduction

The Simple Network Management Protocol (SNMP) is an industry standard for the management and configuration of network components. SNMPv1 is used to configure and monitor the FAF.

Note: Because the FAF stores its configuration in volatile memory, manual changes to the configuration are lost after a reboot. However, the FAF automatically reloads the configuration data from the configuration files and from the Manager (MGR).

5.2 MIB Files

All information that can be configured or viewed with SNMP is described in the Management Information Base (**MIB**) files. The MIB files are located in `/usr/local/share/snmp/mibs`.

The FAF uses the following MIBs:

- `textpass-faf-mib.my`
- `textpass-eia-gen-mib.my`
- `textpass-gen-mib.my`
- `mbalance-mib.my`

5.3 SNMP Manager

For configuration and monitoring purposes, an SMNP Manager or Management Station issues SNMPv1 requests to the FAF. By default, the FAF accepts SNMP requests on UDP port **11461**.

The FAF does not require SNMP requests originated from a specific IP address or UDP port. The FAF requires the SNMP community string to be `public` for get and get-next operations, and `private` for set operations. Otherwise, the FAF silently discards the request.

Note: If '`snmpPropListenAddressType`' parameter in semi-static configuration file is set to 'dual', then FAF will accept requests on both IPv4 and IPv6.

5.4 Trap Service

Up to eight SNMP managers can subscribe to the FAF trap service. When a trapped event occurs, the FAF sends an SNMP trap to all SNMP management stations that are subscribed to the trap service. To subscribe an SNMP manager to the trap service, add an entry to the FAF alarm station table. The entry must contain the IP address (IPv4 or IPv6) or Hostname and UDP port number of the SNMP manager to which SNMP traps should be sent.

The alarm station table is managed through SNMP. Refer to *Configuration* for information about how to configure a trap receiver. Refer to the MIB files for more information about the table.

By default, the FAF originates SNMP traps from UDP port 11462. For traps, the FAF always uses a community string of `public`.

Note:

1. If '`snmpPropAlarmOwnIpv6Address`' parameter in semi-static configuration file is set, then specified address will be used as source address for sending SNMP traps to SNMP Manager with address of type IPv6.
2. If '`snmpPropAlarmOwnIpAddress`' parameter in semi-static configuration file is set, then specified address will be used as source address for sending SNMP traps to SNMP Manager with address of type IPv4.

Refer to the ZephyrTel Mobile Messaging SNMP Trap Reference Guide for the traps that the FAF generates.

Chapter 6

Configuration

Topics:

6.1 Introduction

The FAF has a distributed architecture and central configuration management. The FAF configuration has two parts:

- Semi-static configuration that defines fundamental FAF parameters
- Dynamic configuration that defines FAF filters and filter conditions

Both configuration types are XML-based and are described in this chapter.

6.2 Configuration File Structure

The configuration file is structured as follows:

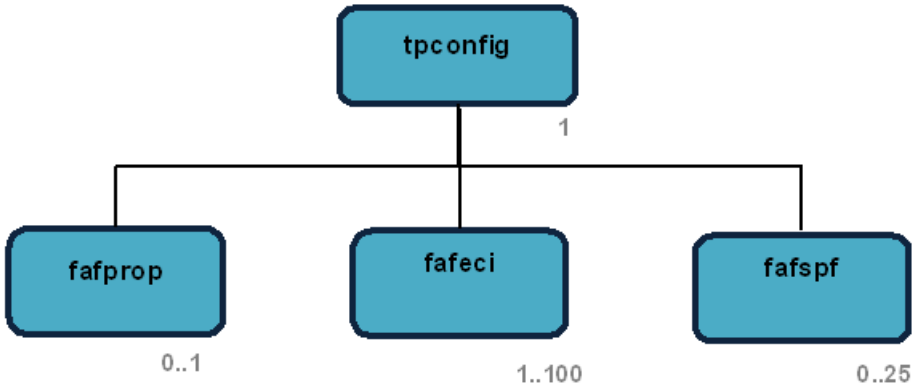


Figure 23: Configuration file structure

6.3 Semi-Static Configuration

The semi-static configuration consists of two files:

- Host-specific configuration file: Contains parameters for a specific FAF and is located at `/usr/TextPass/etc/<hostname>_config.txt`, where `<hostname>` is the host name of the FAF
- Common configuration file: Contains parameters that are common to all FAFs and is located at `/usr/TextPass/etc/common_config.txt`

Configuration parameters can be placed in either file. In case of a conflict in the settings of a parameter, the host-specific configuration file always takes precedence over the common configuration file.

6.3.1 tpconfig Entity

This section describes the `tpconfig` attributes.

6.3.1.1 ipaddress

Mandatory/Optional

Optional

Location

Host-specific configuration file

Description

IP address of the server.

6.3.1.2 runtextfafprocess

Mandatory/Optional

Mandatory (for running the FAF)

Location

Host-specific configuration file

Description

Specifies whether the FAF process should be started. Should be "true" for running the FAF.

Valid Values

- true
- false

Default

false

6.3.1.3 runtextpassprocess

Mandatory/Optional

Mandatory (for running the RTR)

Location

Host-specific configuration file

Description

Specifies if the RTR process should be started. Should be "true" for running the RTR.

Valid Values

- true
- false

6.3.1.4 snmppropalarmownipaddress

Mandatory/Optional

Optional

Location

Common/Host configuration file

Description

The IPv4 address of the TextPass node. If set, this address will be used as source address for sending SNMP traps. This parameter is used to populate Trap Agent address for IPv4/IPv6 address. If this parameter is not set, then the IPv4 address of first network interface is used to populate Trap Agent Address in SNMP Traps.

Valid Value
IPv4 address

Default

Empty string

6.3.1.5 snmppropalarmownipv6address

Mandatory/Optional

Optional

Location

Common/Host configuration file

Description

The IPv6 address or hostname of the TextPass node. If set, this address will be used as source address for sending SNMP traps.

Valid Values

IPv6 address or hostname(maximum length can be of 255 characters)

Default

Empty string

6.3.1.6 snmpproplistenaddresstype

Mandatory/Optional

Optional

Location

Common/Host configuration file

Description

This parameter indicates whether SNMP Listener type is IPv4 only or Dual-stack.

Valid Values

ipv4 or dual

Default

ipv4

6.3.2 fafprop Entity

This section describes the `fafprop` attributes.

6.3.2.1 dupsclustertrapwarningthreshold

Mandatory/Optional

Optional

Location

Common configuration file

Description

Number of duplicates clusters above which the FAF will issue the clustersInUseAlert trap (range 250 to 449). It is recommended that this parameter be placed in the common configuration file.

Valid Values

250 - 449

Default

300

6.3.2.2 enablealternativeduplogic

Mandatory/Optional

Optional

Location

Common configuration file

Description

This parameter is by default set to FALSE. If this parameter is set to TRUE, each instance of a repeating feature is counted and calculated in the similarity.

For example, the original text is featurised to be ABCDEFGH, each of the capital letter is a different feature. A cluster is created for this text. The featurised text ABABABAB will be detected as a duplicate of cluster for ABCDEFGH. This is because all the repeating instances of features A and B are in the cluster, so the similarity is 100% according to the algorithm. If this parameter is FALSE, the repeating feature is only counted once in the calculation. For example, the above featurised text ABABABAB will only have 25% similarity to text ABCDEFGH because A and B are counted only once, and 2 out of the 8 features are present in the two texts.

Valid Values

- true
- false

Default
false

6.3.2.3 enableispiconntcpkeepalive

Mandatory/Optional

Optional

Location

Common configuration file

Description

This parameter is used to control the enabling/disabling of the TCP keep-alive functionality for ISPI connections. When set to true, TCP keep-alive is automatically enabled on every ISPI client session socket started thereafter. When set to false, TCP keep-alive is disabled on every ISPI client session socket.

If the Automatic Blacklisting functionality is supported, it is recommended to set this parameter to true.

Valid Values

•True

•False

Default

False

6.3.2.4 fafenableclearvolumecounterforblacklist

Mandatory/Optional

Optional

Location

Common configuration file

Description

This parameter indicates whether to reset the volume counter or not. When this parameter is set to "true", then FAF will reset the volume counter after successfully adding the blacklist subscriber to SPF. If this parameter is set to "false", the volume counter would not be cleared.

Valid Values

- true
- false

Default
false

6.3.2.5 maxlistlength

Mandatory/Optional

Optional

Location

Common configuration file

Description

Controls the maximum internal memory units that the FAF can use for a list. It is recommended to place this parameter in the common confguration file.

Valid Values

6400 - 64,000 bytes

Default

6400

6.3.2.6 mgrpass

Mandatory/Optional

Optional

Location

Host-specific or Common configuration file

Description

Defines a password to be used to connect to MGR from FAF.

Note: The password should belong to an existing MGR user.

Valid Values

String with the following characteristics:

- Contains at least 2 alphabet letters.
- Contains at least 1 digit.
- Length between 8 and 64 alphanumeric characters.

Default Value

The default is empty. However, this parameter cannot be set empty while configuring it.

6.3.2.7 mgruser

Mandatory/Optional

Optional

Location

Host-specific or Common configuration file

Description

Defines a username to be used to connect to MGR from FAF.

Note: The username should belong to an existing MGR user.

Valid Values

String with length between 1 and 31.

Default Value

The default is empty. However, this parameter cannot be set empty while configuring it.

6.3.2.8 normalisationmap

Mandatory/Optional

Optional

Location

Common configuration file

Description

Defines the mapping for tokenization. This attribute controls the tokenization text pre-processing step, not the normalization step. The mapping may consist of a list of character groups up to 1000 bytes and each character groups is separated by a new line character (`
`). It is recommended to place this parameter in the common configuration file.

Valid Values

String up to 1000 bytes.

Default

```
normalisationmap="0oO&#246;&#214;&#10;1iIlL!\/&#10;2zZ&#10;3eE&#10;4aA&#228;&#196;
&#10;5sS&#223;&#10;6&#10;7tT&#10;8bB&#10;9gG&#10;cC&#10;dD&#10;fF&#10;hH&#10;
jJ&#10;kK&#10;mM&#10;nN&#10;pP&#10;Q&#10;rR&#10;uU&#252;&#220;&#10;vV&#10;
wW&#10;xX&#10;yY"
```

6.3.2.9 wordboundary

Mandatory/Optional

Optional

Location

Common configuration file

Description

Defines a customized word boundary set. When not provisioned, the default word boundary set will be used. The control characters in ASCII and extended ASCII do not need to be provisioned.

Valid Values

String up to 1000 bytes.

Default

The following are the default word boundaries:

- Control characters of ASCII and extended ASCII set (`0x00,0x1f` and `0x7f,0x9f`]
- White space

- `~`!@#$%^&*()+-_=[]{}\|:";'<>?,./`

6.3.2.10 xmlretrycount

Mandatory/Optional

Optional

Location

Host-specific or Common configuration file

Description

Max number of retries for GT/Network ABL provisioning requests.

Valid Values

1 - 100

Default Value

5

6.3.2.11 xmlretryinterval

Mandatory/Optional

Optional

Location

Host-specific or Common configuration file

Description

Retry interval in seconds for GT/Network ABL provisioning requests.

Valid Values

1 - 600

Default Value

60

6.3.3 fafspf Entity

This section describes the `fafspf` attributes.

6.3.3.1 ipaddress

Mandatory/Optional

Mandatory

Location

Common configuration file

Description

This field indicates the IPv4 address of the SPF server to which the FAF is to connect

Default

127.0.0.1

6.3.3.2 name

Mandatory/Optional

Mandatory

Location

Common configuration file

Description

This field indicates the unique name of the SPF (Core) server to which FAF is to connect. Non-unique names are rejected.

Valid Values

Any string of 1 to 31 UTF-8 bytes.

Default

N/A

6.3.3.3 port

Mandatory/Optional

Optional

Location

Common configuration file

Description

This field indicates the TCP port of the SPF server to which the FAF is to connect.

Valid Values

0 - 65535

Default

9800

6.3.4 fafeci Entity

This section describes the `fafeci` attributes.

6.3.4.1 host

Mandatory/Optional

Mandatory

Location

Common configuration file

Description

IP address of the FWL to which to connect.

6.3.4.2 port

Mandatory/Optional

Optional

Location

Common configuration file

Description

TCP port number to which to connect.

Valid Values

0 - 65535

Default

9500

6.3.4.3 user

Mandatory/Optional

Mandatory

Location

Common configuration file

Description

Log-in user identification; must match user parameter of the external condition (EC) application on the FWL.

Valid Values

Maximum 80 characters

6.3.4.4 pass

Mandatory/Optional

Mandatory

Location

Common configuration file

Description

Password; must match password parameter of the EC application on the FWL. Maximum length is 80 characters.

6.3.4.5 inactivitydisconnect

Mandatory/Optional

Optional

Location

Common configuration file

Description

After this interval of inactivity, the connection is dropped. It should be larger than inactivityping if not set to zero. A value of zero means that the FAF will never disconnect.

Valid Values

0 - 3600

Default

0

6.3.4.6 inactivityping

Mandatory/Optional

Optional

Location

Common configuration file

Description

After this interval of inactivity, an ECI lifecheck is sent. A value of zero means that an ECI lifecheck is never sent.

Valid Values

0 - 3600

Default

5

6.3.5 trapreceiver Entity

This section describes the `trapreceiver` attributes.

6.3.5.1 ipaddress

Mandatory/Optional

Mandatory

Location

Host-specific or Common configuration file

Description

IP address (IPv4 or IPv6) or Hostname of the trap receiver.

6.3.5.2 udpport

Mandatory/Optional

Optional

Location

Host-specific or Common configuration file

Description

UDP port on the trap receiver to which traps are sent.

6.4 Dynamic Configuration

The dynamic configuration is called dynamic because, in general, the parameters change frequently. The dynamic configuration is configured in the MGR, which is a Web interface.

6.4.1 Create an Advanced Filter

To create an advanced filter:

1. In the left navigation bar, select **Advanced Filters ➤ Filters**.
   The Advanced Filters tab appears.
2. Click **Add New**.
   A new Advanced Filters tab appears.
3. Enter a unique name for the filter in the **Name** box (up to 31 characters).
4. Optionally enter a description of the filter in the **Description** box.
5. Enter a filter priority between 0 and 100 in the **Priority** box (defaults to 50).
   Filters with a higher priority are evaluated first.

   Note: The priority between -100 to 0 shall no longer be used. Rearrange the priority to the [0, 100] range if negative priority is used.

6. From the **Action** list, select the action that the RTR should take if a message meets all conditions when the FAF processes the filter:

   - Return True—The FAF returns true for the message fields
   - Return False—The FAF returns false for the message fields
   - Continue—The FAF should continue to process the next filter

   When first creating the filter, select "Return True". Then, after you create and activate the desired conditions for the filter, change the filter action to "Return False". If you create a filter with no conditions and a "Return False" action, the FAF will immediately return "false" to the RTR and will therefore block messages.

7. In **Blacklist Party** field, select which subscriber should be blacklisted.

   - None: Indicates that no ABL is configured for this filter.
   - Originator: Originator (i.e. A-party subscriber) will be blacklisted
   - Recipient: Recipient (i.e. B-party subscriber) will be blacklisted
   - Originating GT: SMSC GT will be blacklisted
   - Originating Network: SMSC Network will be blacklisted

   Default value is 'None'.

   If Blacklist Party is 'None', then continue at *Step 11*.

8. If Blacklist Party field is selected as "Originator" or "Recipient", select the Auto Blacklist service, which will be assigned to blacklisted subscriber, in the **Originator Blacklist Service** or the **Recipient Blacklist Service** field (whichever is displayed).

   - Originator Blacklist Service: If Blacklist Party is 'Originator', then only Originator Auto Blacklist services (i.e. ABL services created with the "Invoking Address" set to "Originator") will be shown in drop-down list. Select the appropriate one.
   - Recipient Blacklist Service: If Blacklist Party is 'Recipient', then only Recipient Auto Blacklist services (i.e. ABL services created with the "Invoking Address" set to "Recipient") will be shown in the drop-down list. Select the appropriate one.

If Blacklist Party field is selected as "Originating GT" or "Originating Network", select the ABL List, in which blacklisted GT or Network will be added.

- List: If Blacklist Party is "Originating GT" or "Originating Network", then only the lists for which ABL list checkbox is checked will be shown in the drop-down list. Select the appropriate one.

Note: For the procedure to create an Originator/Recipient Auto Blacklist service, refer to the MGR Operator Manual.

Note: For the procedure to create a List, refer to the MGR Operator Manual.

Important: DO NOT configure the same Originator/Recipient Blacklist Service on multiple filters, if their Action is set to "Continue". Doing so is likely to lead to unpredictable system behavior with respect to the Automatic Blacklisting functionality.

9. Select the blacklist blocking action type in **Blacklist Action** field.

   - Permanent Blocking: Subscriber will be blocked permanently.
   - Time-based Blocking: Subscriber will be blocked for the duration specified in Blacklist Duration field.
   - Absolute Blocking: SMSC GT or Network is to be blacklisted until a specific date time. This option is available only when the Filter Blacklist Party is selected as "Originating GT" or "Originating Network".

   If Blacklist Action is Permanent Blocking, then continue at *Step 11*.

10. If Blacklist Action is Time-based Blocking, the **Blacklist Duration** field will be displayed. In **Blacklist Duration** field, indicates the time-duration for which subscriber will be blacklisted.

    The Blacklist Duration should be less than 99 days (Max allowed duration is 98 Days 23 Hours 59 Minutes). Minimum allowed Blacklist Duration is 1 Minute.

    If the Blacklist Action is Absolute Blocking, the **End Date** field will be displayed.

    The filter's End Date specifies the last date time for which subscriber or network should be blacklisted.

    This parameter is relevant only if the filter's Blacklist Action is set to "Absolute Blocking".

11. In the **Append** box, optionally enter any text that the FAF should append to the message. The FAF will append this text if the message meets all conditions of the filter and if the Data message field was provided to the FAF.

    CAUTION: Do not use text replacement/append functions that may make the user data longer than the original user data. When applied to a "full segment" this will lead to an undeliverable message.

12. Click **Save**.

    The MGR saves the filter and closes the tab.

13. Activate the filter.

6.4.2 Add Conditions to an Advanced Filter

Prerequisites:

- Filter

- Filter list (if adding a content condition)

Combine advanced filters and conditions of different types to create filter conditions.

To add a filter condition to a filter:

1. In the left navigation bar, select **Advanced Filters ➤ Filters**.
   The Advanced Filters tab appears.
2. Click the name of an existing filter.
3. In the Filter Conditions section, click **Add New**. A Filter Conditions tab appears.
4. If the filter condition should be inverted if the condition is true, select **Invert**.
5. From the **Filter Name** list, select the filter to use (defaults to the filter that you clicked in the Advanced Filters tab).
6. In the **Name** box, enter the name of the filter condition.
7. From the **Type** list, select the condition type.
8. Click **Save**.

   The MGR creates the filter condition and closes the tab.
9. Activate the filter condition.

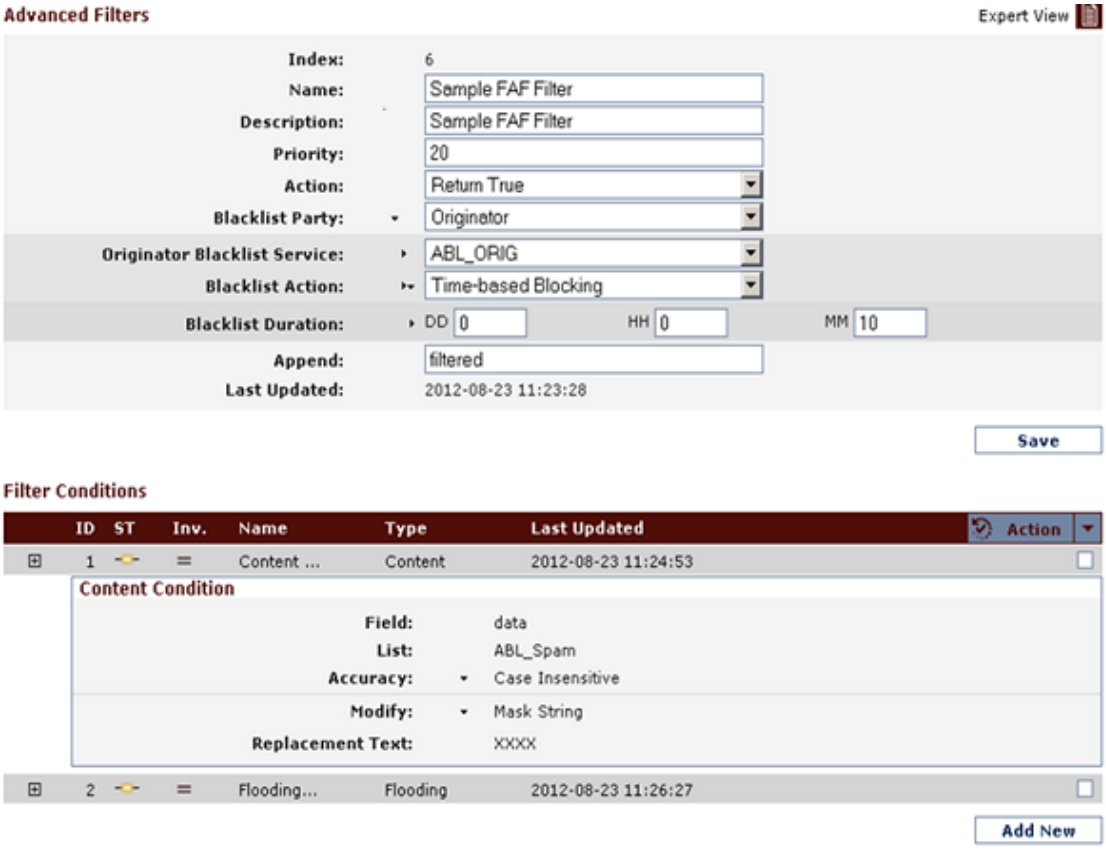


Figure 24: Sample filter with conditions

6.4.2.1 Add a Content Condition

When adding a content condition:

1. From the **Field** list, select the message field to which the condition should be applied; the default and most commonly used field is Data (message content).

   Refer to *Message Field Options* for a complete list of available message fields and descriptions.

2. From the **List** list, select the advanced filter list to use.
3. From the **Accuracy** list, select the accuracy level for matching:

   - Exact
   - Case-insensitive
   - Tokenised
   - Normalised
   - Regular Expression

   The accuracy indicates an implicit transformation that the FAF performs on all text involved in the match before the match is calculated.

   Note: The case-insensitive match only works for characters that are supported by operating system function call `tolower()`. In most operating systems, the `tolower()` function supports basic ASCII. Some operating systems also support extended ASCII that includes German and Nordic characters. You can verify the characters supported by the operating system `tolower()` function with the following command on the command line interface:

   ```
   # locale -k LC_CTYPE
   ```

4. In the **Whole Words Match** checkbox specify if the FAF should perform matching on whole words only. For example:

   - Exact matching: `"apple"` matches text `"this is an apple."`, but not `"this is a pineapple."` or `"these are apples."`.
   - Case-insensitive matching: `"apple"` matches `"Apple is good."`, but not `"PineApple is good."`.

   Note: This checkbox is only available for the Exact and Case-insensitive **Accuracy** matches.

5. From the **Modify** list, select how the target text should be modified:

   - None
   - Mask string (will not increase the length of the target string)
   - Replace string (may affect the length of the target string)
   - Replace message

   CAUTION: Do not use text replacement/append functions that may make the user data longer than the original user data. When applied to a "full segment" this will lead to an undeliverable message.

6. If you select a content modification option other than **None**, enter the text that should mask or replace the target text in the **Replacement Text** box.

   - Note that the **Modify** option is relevant only for the user data field and should be set to **None** for all other message fields.

6.4.2.1.1 Message Field Options

The message fields that can be selected in **Field** list for the content, duplicates, flooding, and bulk conditions:

Field (number)	ECI Field	Description	Additional Information
path (1)	routingPath	Routing path	Integer format. Supported values are: • moMo (0) • moMt (1) • moMtMo (2) • moMtAt (3) • moAt (4) • moDiscardWithNack (5) • moDiscardWithAck (6) • moDiscardSilently (7) • mtMt (10) • mtBlockWithTemporaryError (11) • mtBlockWithPermantError (12) • mtBlockWithNoResponse (13) • mtBlockWithAck (14) • aoAo (20) • aoMt (21) • aoMtAo (22) • aoAt (23) • aoDiscardWithAck (24) • aoDiscardWithNak (25) • atAt (30) • atBlockWithTemporaryError (31) • atBlockWithPermanentError (32) • atBlockWithAck (33)
submit (3)	originalSubmitTime	Not adjusted submit time	Original submission time, in Unix time format.
uniq (4)	uniqueSubmitTime	Adjusted (made unique) submit time	Time the message was submitted to the RTR. Note that this field contains the potentially adjusted submission time as described in the RTR Operator Manual chapter on Service Center Time Stamps.
deliv (5)	deliveryTime	Delivery time	Time the RTR delivered or deleted the message. In Unix time format.
orig (6)	originatorAddress	Originator address	Self explanatory. In ASCII string format with prefix. National number is with prefix "N", international is with prefix "+", unknown is with prefix "U", alphanumeric is with prefix "A".

Field (number)	ECI Field	Description	Additional Information
			For example, N12345678, Aalphanumber.
origImsi (7)	originatorImsi	Originator IMSI	Self explanatory. In ASCII string format.
smsc (8)	smscAddress	SMSC address	Self explanatory. In ASCII string format.
msc (9)	mscAddress	MSC address	Self explanatory. In ASCII string format.
recip (10)	recipientAddress	Recipient address	Self explanatory. An ASCII string format with the same prefix as "orig".
recipImsi (11)	recipientImsi	Recipient IMSI	Self explanatory. In ASCII string format.
segTotal (12)	cmTotalSegments	Total number of segments	Total number (0-255) that indicates the total number of pieces in a concatenated message (only present in case of a concatenated message when received as a part of SMPP `sar_total_segments` or as a part of 8 bit reference or 16 bit reference number UDH IEI).
segId (13)	cmCurrentSegment	Segment number	Current segment number (0-255) of the concatenated message. A running number for each part of a concatenated message (only present in case of a concatenated message when received as a part of SMPP `sar_segment_seqnum` or as a part of 8 bit reference or 16 bit reference number UDH IEI).
len (15)	lengthOfMessage	Length of message	Number of characters in septets or octets, depending on the data coding scheme (DCS).
header (17)	userDataHeader	User data header	Value specified in one of the information element identifiers (IEIs) of the user data header (UDH) of the message. Only evaluates positively if there is an exact match with the information element identifier (00-FF, hexadecimal) of the UDH. Refer to technical specification 3GPP 23.040 for more information. Most common IEI values: • 00: Concatenated short message

Field (number)	ECI Field	Description	Additional Information
			• 01: Special SMS message indication • 04: Application port addressing scheme, 8-bit address • 05: Application port addressing scheme, 16-bit address • 06: SMSC control parameters • 07: UDH Source Indicator
encoding (18)	dataCodingScheme	Data Coding Scheme (DCS)	Data coding scheme (DCS) specified in the message. The value is in decimal format.
data (19)	userData	Message data	The content can be already modified by other filters. Decoded into UTF-8 format.
statrepinfo (20)	statusReportInfo	Status report information	Opaque value containing the reference for the SS7 Status Report assigned by the TPR.
smppMsgId (21)	smppMessageId	SMPP message ID	In string format.
dataorg (22)	userData	User data	This is the original message content from ECI. Decoded into UTF-8 format.
segRef (23)	cmReferenceNr	Segment reference	Same for all segments in a concatenated SMS (received as a part of SMPP `sar_msg_ref_num` or as a part of 8 bit reference or 16 bit reference number UDH IEI).
notifreq (24)	notificationRequest	Notification request	Indicates whether notification or status report was requested for this message (true) or not (false).
callingGt (25)	callingPartyAddress	Calling global title	GT of the SCCP Calling Party Address in the message. In ASCII string format.
calledGt (26)	calledPartyAddress	Called global title	GT of the SCCP Called Party Address in the message. In ASCII string format.
delstat (42)	deliveryStatus	Delivery status	Result of a delivery attempt as reported in a notification on an AO/SM. In integer format. Supported values are: • noStatusAvailable (0) • inProgress (1) • validityPeriodExpired (2) • deliveryFailed (3)

Field (number)	ECI Field	Description	Additional Information
			• deliverySuccessful (4) • noResponse (5) • lastNoResponse (6) • cancelled (7) • deleted (8) • deletedByCancel (9) • scheduled (10) • accepted (11) • rejected (12) • skipped (13) • replaced (14)
exconrule (55)	selectedExternalConditionRule	Name of external condition rule	Name of the external condition rule used to forward the message to the FAF.
protocolId (60)	protocolId	Protocol identifier	Indicates the value of the TP-PID field (included in the MAP header), if the message was MO or MT. Otherwise (i.e. for AO or AT messages) it indicates the value of the protocol id parameter included (if any) in the message. This field is not applicable for status reports and notifications. Valid values are in the range 0-255.

Note: The 'protocolId' message field should not be selected while configuring a duplicates, flooding or bulk filter condition, because it has no relevant use case for these filters.

6.4.2.1.2 Accuracy Options

The message text is matched with other text with a particular "accuracy". The accuracy indicates an implicit transformation that is performed on all text involved in the match, before the match is calculated.

The following options are available for the **Accuracy** parameter.

Option (Short Name)	Description
Exact (exact)	The target text must contain an exact match to an entry in the specified list. No transformation is done. For example: • `hello` matches `hello` • `hEllo` does not match `hello`

Option (Short Name)	Description
Case Insensitive (case)	The FAF removes the case from all characters (all characters are effectively lowercased) of the target text before attempting to match it to an entry in the specified list. For example: • `Hello` matches `hello` • `H3llo` does not match `hello` The FAF only supports case removal for the ASCII character set. Therefore, it is not recommended to use this option in combination with non-ASCII characters.
Tokenised (token)	The target text and all entries in the specified list are tokenised before the FAF attempts to match the text to an entry in the list. For example (assuming the default normalisation map is in use): • `H3ll0` matches `hello` • `HH3ll0` does not match `hello`
Normalised (repeat)	The target text and all entries in the specified list are normalised before the FAF attempts to match the text to an entry in the list. For example (assuming the default normalisation map is in use): • `H33ll1000` matches `hello` • `hollo` does not match `hello`
Regular Expression (regexp)	The target text is matched using regular expression statements that are provisioned by a list. Any non-regular expression strings from the list are ignored. For example: • `hello` matches `/(hello|hi|hoi)/` • `hello` does not match `/[0-9]/` • `hello1234` matches `/[0-9]+/` The regular expression conforms to the POSIX Extended Regular Expression standard. The FAF uses the POSIX regular expression library. Therefore, the limitation of the library also applies to the FAF. It is recommended to use basic regular expression grammar. **Note:** To support regular expressions, the FAF requires that the `en_US.UTF-8` locale is installed. Refer to the Full Element Installation Manual for more information.

6.4.2.1.3 Modify Options

The following options are available for the **Modify** parameter.

Option	Description
None	The FAF does not modify the target text.

Mask String	The **Mask String** functionality ensures that the replacement of the text does not increase the length of the original message. This prevents truncation of the message.
	For example, there is a content filter for the word `website` and the **Replacement Text** is `CENSORED`. If the message is:
	`This website is good`
	The message will be modified to:
	`This CENSORE is good`
	In this example, the message length remains the same.
	If the **Replacement Text** is `***`, the message will be modified to:
	`This *** is good`
	In this example, the message length decreases.

Replace String	The matching string of the target text is replaced by the text in the **Replacement Text** box. As a result, the modified target text may increase or decrease in length.
Replace Message	The entire target text is replaced by the text in the **Replacement Text** box.

6.4.2.2 Add a Duplicates Condition

When adding a duplicates condition:

Note: Certain parameter changes in the duplicates filter condition may take a long time to effect due to the large state that is kept in the filters. Especially when the filter is full (has the maximum amount of memory state), it can take quite some time for the changes to take effect (sometimes 30 minutes or more).

1. From the **Field** list, select the message field to evaluate; the default and most commonly used field is Data.

   Refer to *Message Field Options* for a complete list of available message fields and descriptions.

2. In the **Spacing** box, enter the number of allowed dissimilar messages (2-99,999) between two similar messages (default 1,000).

   If more messages than this are found between two similar messages, the similar messages are not considered duplicates.

3. In the **Min. Size** box, enter the minimum number of messages (2-1,000) required to start a duplicates cluster (defaults to 10).
4. In the **Length** box, enter the minimum number of features (4-160) required for to start duplicates clustering (defaults to 4).
5. In the **Threshold** box, enter the minimum number of messages (2-999,999) that must be in the cluster for the duplicates condition to return true (defaults to 10).
6. In the **Similarity** box, enter the percentage (0-100) that messages must be similar to be placed in the same cluster (defaults to 80%).

100% means the messages must match exactly.

7. In the **Delete Age** box, enter the number of seconds (0-999,999) that a never-matched cluster is allowed to exist before it is deleted (0 means clusters are never deleted).

   If the cluster is matched, the timer restarts for the cluster.

6.4.2.3 Add a Flooding Condition

When adding a flooding condition:

1. From the **Field** list, select the message field to evaluate.

   The field should depend on the type of traffic that is being evaluated for flooding. For example, for MO traffic, the MSC should be used; for MT traffic, the SMSC should be used.

   Refer to *Message Field Options* for a complete list of available message fields and descriptions.

2. In the **Significant Digits** box, enter the number of digits (0-16) of the specified field that are taken into account when tracking originators (defaults to 16, which is recommended).

   The first digits of the number are used; for example, for a setting of `6`, the originator numbers `15562235` and `155622234` map to the same trackable originator.

   More digits may be taken into account if the FAF detects higher traffic.

3. In the **Minimal Traffic** box, enter the minimum number of messages per second (1-1,000,000) required for the filter to become active (defaults to 5, which is recommended).

   This is the constant threshold used to compute the flooding detection threshold. Use this setting to prevent spurious flooding detection at low traffic levels. Higher values make flooding detection less likely.

4. In the **Traffic Increase Rate** box, enter a relative increase in traffic (1-10,000%) used to compute the flooding detection threshold.

   This is the traffic increase that is required to trigger the filter. The value is relative to the long-term traffic average. Higher values make flooding detection less likely.

5. In the **Time Delay** box, enter the number of seconds (1-10,000) that the short-term traffic average must exceed the flooding detection threshold before the filter becomes active (defaults to 30).

   The condition only yields true this number of seconds after the flooding detection threshold has been exceeded. Higher values make flooding detection less likely.

6. In the **Filter Period Flooding** box, enter the number of seconds (1-10,000) to use to calculate the short-term traffic average (default 10).

   This is the response time used to compute the short-term traffic average. Fluctuations shorter than this period are filtered out. Higher values make flooding detection less likely.

7. In the **Filter Period Baseline Traffic** box, enter the number of seconds (1-10,000) to use when calculating the long-term traffic average (defaults to 3600).

   Fluctuations slower than this are not considered flooding. This value should be significantly higher than the **Time Delay**; a factor of at least 20 is recommended. Higher values make flooding detection less likely.

8. In the **Margin** box, enter the threshold, in messages per 1000 seconds (1-100,000), below which traffic from a trackable originator is not tracked (default 5).

For example, for a value of 10, originators sending less than 10 messages per 1000 seconds are not tracked. Use this parameter to avoid using FAF processing performance for message originators for which the traffic lies below a level of interest. Higher values make flooding detection less likely.

6.4.2.4 Add a Volume Condition

When adding a volume condition:

1. From the **Group By** list, select a value that determines on which fields to group and count messages for the Volume filter condition.

   Possible options are:

   - `Nothing` — Count all messages
   - `Originator` — Group and count messages based on originator address (as specified in the TP-Originating-Address (TP-OA) field of the SMS Message).
   - `Content` — Group and count messages based on raw SMS user data content (as specified in the TP-User Data (TP-UD) field of the SMS Message).
   - `Originator + Content` — Group and count messages based on originator address and raw SMS user data content

2. In the **Memory** field, enter the amount of memory in Megabytes (MB) to dedicate for the tracking of elements in the filter. Valid values range from 256 to 65,536 MB. Default is 1024 MB.

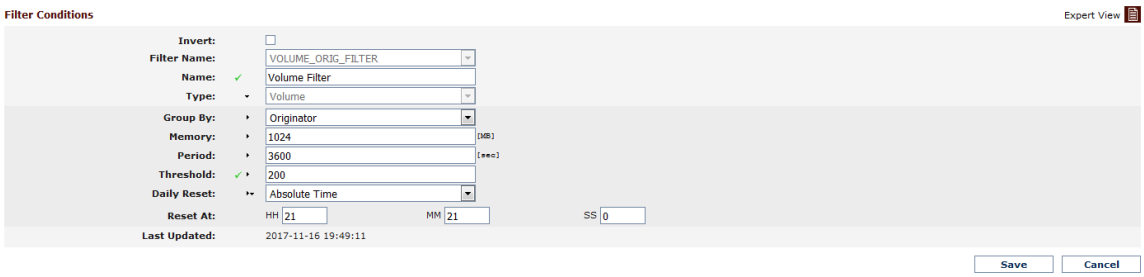
   This value is a hard memory limit on the amount of memory dedicated for storing data for this filter. When the memory limit is reached, a trap is generated.

   Refer to the FAF Operator Manual, Volume Condition for more information on memory dimensioning.

3. In the **Period** field, enter the tracked period of time in seconds. Valid values range from 60 seconds (one minute) to 86,400 seconds (one day) . Default is 3600 seconds (one hour).
4. In the **Threshold** field, enter the number of messages in each grouping after which the condition shall apply (return 'true'). Valid values range from 0 to 2,147,483,647 messages. Default is 200 messages.
5. In the **Daily Reset** field, select **Absolute Time** to reset the volume counter for the subscriber. The default value is **None**, meaning that no volume counter will be reset.
6. In the **Reset At** field, enter the hours, minutes and seconds (default 0, 0, 0 respectively). This is the absolute time when the volume counter for subscriber will be reset. This field is applicable when **Daily Reset** is set to **Absolute Time**.

   The interval format is HH MM SS, where:

   - HH is the hour, 0-23
   - MM is Minute, 0-59
   - SS is Second, 0-59



Note: While doing configuration of this field from `tp_shell`, the user needs to provide the duration (HHMMSS) in `fafVolumeDailyResetTimeHour`, `fafVolumeDailyResetTimeMinute` and `fafVolumeDailyResetTimeSecond` fields.

6.4.2.5 Add a Bulk Condition

When adding a bulk condition:

Note: Certain parameter changes in the bulk filter condition may take a long time to effect due to the large state that is kept in the filters. Especially when the filter is full (has the maximum amount of memory state), it can take quite some time for the changes to take effect (sometimes 30 minutes or more).

1. From the **Field** list, select the message field to evaluate; the default and most commonly used field is Orig.

   Refer to *Message Field Options* for a complete list of available message fields and descriptions.

2. In the **Threshold** box, enter the minimum number of seconds (0-999,999) to mark a message as bulk (defaults to 10).

   When the average time span between messages is lower than this threshold, the message is regarded as bulk and the condition returns false.

3. In the **Window Size** box, enter the window size (0-999,999) used to calculate the average timespan (defaults to 64).

   This is the filter constant for the auto-regressive low-pass filter, which is based on the following algorithm:

   ```
   new_value = old_value * C + window_size * (1 - C)
   ```

   Where:

   - `window_size` is the timepan between two subsequent messages
   - `window` is this parameter
   - `C = exp(-timespan/window)`

   The larger the window size, the slower the low-pass filter becomes; that is, the less responsive the filter becomes to quick changes in value.

4. In the **Expiration Period** box number of seconds (0-999,999) before a record expires (default 3600).

   When a record matches, the FAF updates the record's timestamp. When the timestamp is older than the expiration time, the FAF deletes the record.

6.4.2.6 Add an Enhanced Messaging (EMS) Condition

When adding an enhanced messaging (EMS) condition:

1. In the **Protocol Id Values** box, enter the desired protocol id value(s) against which the EMS filter condition should compare the protocol id of a message. If no protocol id value is entered, then the filter condition will match any message protocol id.
2. Select one of more Information Element IDs (IEIs) on which to filter:

   - 00 - Concatenated short messages, 8-bit reference number
   - 01 - Special SMS Message Indication
   - 03 - Value not used to avoid misinterpretation as LF character
   - 04 - Application port addressing scheme, 8 bit address
   - 05 - Application port addressing scheme, 16 bit address
   - 06 - SMSC Control Parameters
   - 07 - UDH Source Indicator
   - 08 - Concatenated short message, 16-bit reference number
   - 09 - Wireless Control Message Protocol
   - 10 - Text Formatting
   - 11 - Predefined Sound
   - 12 - User Defined Sound (iMelody max 128 bytes)
   - 13 - Predefined Animation
   - 14 - Large Animation (16*16 times 4=128 bytes)
   - 15 - Small Animation (8*8 times 4 = 8*4 = 32 bytes)
   - 16 - Large Picture (32*32 = 128 bytes)
   - 17 - Small Picture (16*16 = 32 bytes)
   - 18 - Variable Picture
   - 19 - User prompt indicator
   - 20 - Extended Object
   - 21 - Reused Extended Object
   - 22 - Compression Control
   - 23 - Object Distribution Indicator
   - 24 - Standard WVG Object
   - 25 - Character Size WVG Object
   - 26 - Extended Object Data Request Command
   - 32 - RFC 822 E-Mail Header
   - 33 - Hyperlink format element
   - 34 - Reply Address Element
   - 35 - Enhanced Voice Mail Information
   - 36 - National Language Single Shift
   - 37 - National Language Locking Shift

   If the user data header (UDH) of a message contains the selected IEI(s), FAF will return "true" for the condition. The EMS condition uses a logical OR operation; therefore, if you select multiple IEIs and also configure certain protocol id values for one EMS condition and the message contains any of the selected IEIs OR any of the configured protocol ids, the condition will return "true". Refer to the 3GPP 23.040-920 specification for a description of EMS IEIs.

3. If you selected "04 - Application port addressing scheme, 8 bit address":
   a. Enter an 8-bit source port number in the **8 Bit Source Port** box.
   b. Enter an 8-bit destination port number in the **8 Bit Destination Port** box.

   If the source port and/or destination port in the UDH of a message matches the provisioned port, FAF will return "true" for the condition. If you do not provision port numbers, FAF simply checks for the presence of the IEI, and returns "true" if it is present.

4. If you selected "05 - Application port addressing scheme, 16 bit address":
   a. Enter a 16-bit source port number in the **16 Bit Source Port** box.
   b. Enter a 16-bit destination port number in the **16 Bit Destination Port** box.

   If the source port and/or destination port in the UDH of a message matches the provisioned port, FAF will return "true" for the condition. If you do not provision port numbers, FAF simply checks for the presence of the IEI, and returns "true" if it is present.

6.4.2.7 Add an Expression Condition

When adding an expression condition:

1. In the **Test Expression** box, enter the expression to test:
   Below there are examples of expressions:

```
messagetype == 4
```

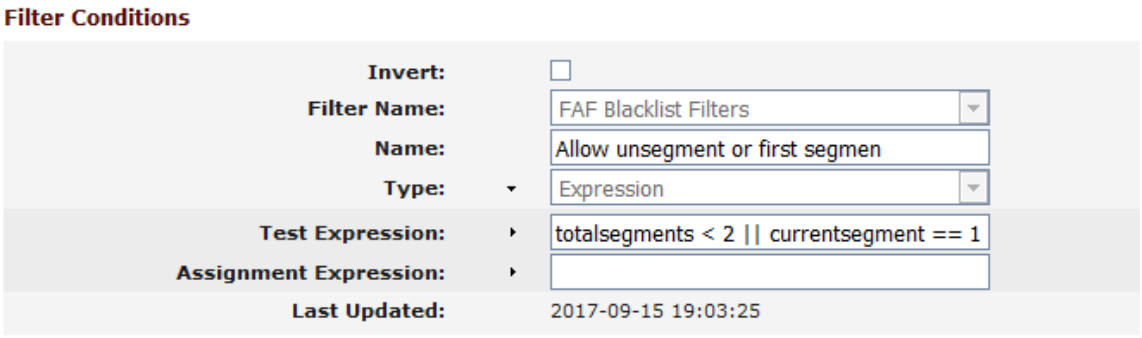
```
totalsegments >= 4
```

```
messagetype == 0 && totalsegments >= 4
```

   In case of a concatenated SM, it is counted by message, not by segment. So, in concatenated message counted only the first segment of a concatenated message. Expression condition is used to filter the segments of SM. In this configuration, expression condition must be configured in the filter condition.

   For example:

```
totalsegments < 2 || currentsegements == 1
```



2. In the **Assignment Expression** box, enter the expression to assign a value to a variable.
   For example:

```
eciattribute2 = 1
```

6.4.2.7.1 Expression Variables

The following variables are available for use in expression conditions:

Variable	Valid Values	Can be used in...
`messagetype`	• 0: MO short message • 1: MT short message • 2: AO short message • 3: AT short message • 4: HLR SRI-SM request • 5: HLR SRI-SM response • 6: MT delivery notification • 7: AT delivery notification	Test expression
`failuremessagekey`	An integer	Test expression and assignment expression
`eciattribute[n]` where `[n]` is 1 through 32	0 or 1	Test expression and assignment expression
`totalsegments`	An integer. Range - 0 to 255 `totalsegments` will have the value of 0 in case the message is not segmented.	Test expression

Note: The test expression containing the expression variable "`totalsegments`" will be evaluated against the value received for the ECI message field "`cmTotalSegments`" in the ECI validation request. If "`cmTotalSegments`" is not included for the ECI application, the "`totalsegments`" expression will be assigned the value of 0.

By default, the ECI message field "`cmTotalSegments`" is not included for the ECI application.

6.4.2.7.2 Expression Operators

The expression condition supports the following operators:

Operator	Meaning
||	Logical OR
&&	Logical AND
<	Less than
<=	Less than or equal to
>	Greater than
>=	Greater than or equal to
==	Equal to
!=	Not equal to

Operator	Meaning
+	Plus
-	Minus
*	Multiplied by
/	Divided by
%	Modulo (remainder of division)

6.4.2.8 Add a Spread Condition

When adding a spread condition:

1. In the **Max. Subscribers** box, enter the maximum number of subscriber numbers the spread filter can contain. Entries are replaced on a least recently used (LRU) basis. This variable can only be changed when the admin state is `inactive`. The valid range is 1-64800000 and the default value is 864000.
2. In the **Recipient Count** box, enter the size of the recipient bitmap. The recipients are hashed via hash function to an entry in the recipient bitmap. This variable can only be changed when the admin state is `inactive`. The valid range is 0-99999 and the default value is 1000. It is recommended to set this to at least thrice the size of the **Recipient Limit**.
3. In the **Search Limit** box, enter the number of records to search for a free slot in the maintained subscriber list by the Spread filter where the new record can be inserted. The starting point of the search in the list will be the index calculated on the basis of hash value of the Originator contained in the message. Increasing this value will increase the accuracy of search in the subscriber list but will also increase CPU time for the spread filter. This variable can only be changed when the admin state is `inactive`. The valid range is 1-999 and the default value is 16.
4. In the **Message Limit** box, enter the limit of number of messages per subscriber. If the number of total sent messages exceeds this value, the number of entries in the recipient bitmap is counted. This variable can be changed when the admin state is `active`. The valid range is 0-2147483647 and the default value is 1000.
5. In the **Recipient Limit** box, enter the limit of different entries in the recipient bitmap that approximates the number of different recipients the originator has sent their messages to. If both this limit and the Message Limit are exceeded, then the spread filter returns true. This variable can be changed when admin state is active. Valid range is (0-9999) and the default value is 300.
6. From the **Reset At** field, select **Daily** to reset the spread filter daily, select **Weekly** to reset the spread filter weekly, or select **Monthly** to reset the spread filter monthly. Default is **Daily**.
7. From the **Day of the Week** field, select a day of week to reset the spread filter weekly. Default is **Monday**. This field is applicable only if **Reset At** is set as **Weekly**.
8. From the **Day of the Month** field, select a day of month to reset the spread filter monthly. Default is 1. This field is applicable only when **Reset At** is set to **Monthly**.
9. In the **Reset Time** field, enter the hours, minutes and seconds (default 0, 0, 0 respectively). This is the absolute time when the spread filter will be reset.

   The interval format is `HH MM SS`, where:

   - `HH` is the hour, 0-23
   - `MM` is the minute, 0-59

- SS is the second, 0-59

Note: While doing the configuration of this field from tp_shell, the user needs to provide the duration (HHMMSS) in the fields fafSpreadResetTimeHour, fafSpreadResetTimeMinute and fafSpreadResetTimeSecond.

6.4.2.9 Add a Delta Condition

When adding a delta condition:

1. In the **Max. Subscribers** box, enter the maximum number of subscriber numbers the delta filter can contain. Entries are replaced on a least recently used (LRU) basis. This variable can only be changed when the admin state is inactive. The valid range is 1-64800000 and the default value is 864000.
2. In the **Search Limit** box, enter the number of records to search for a free slot in the maintained subscriber list by the Delta filter where the new record can be inserted. The starting point of the search in the list will be the index calculated on the basis of hash value of the MSISDN contained in the message. Increasing this value will increase the accuracy of search in the subscriber list but will also increase CPU time for the delta filter. This variable can only be changed when the dmin state is inactive. The valid range is 1-999 and the default value is 16.
3. In the **Recv Send Percent** box, enter the ratio percentage between SMS received and SMS sent by a subscriber. If the ratio percentage is equal to or lesser than the configured value of this field and the total number of messages for that subscriber reaches a value equal to or greater than the value of configured **Message Limit**, the filter matches. This variable can be changed when the admin state is active. The valid range is 0-100 and the default value is 5.
4. In the **Message Limit** box, enter limit of number of messages per subscriber. If the number of total messages (send + receive) is equal to or above this value and the ratio percentage between SMS received and SMS sent is lower than or equal to the configured value of field **Recv Send Percent**, the Delta filter returns true. This variable can be changed when the admin state is active. The valid range is 0-2147483647 and the default value is 1000.
5. From the **Reset At** field, select **Daily** to reset the delta filter daily, select **Weekly** to reset the delta filter weekly, or select **Monthly** to reset the delta filter monthly.
6. From the **Day of the Week** field, select a day of week to reset the delta filter weekly. This field is applicable only if **Reset At** is set as **Weekly**.
7. From the **Day of the Month** field, select a day of month to reset the delta filter monthly. This field is applicable only when **Reset At** is set to **Monthly**.
8. In the **Reset Time** field, enter the hours, minutes and seconds (default 0, 0, 0 respectively). This is the absolute time when the delta filter will be reset.

   The interval format is HH MM SS, where:

   - HH is the hour, 0-23
   - MM is the minute, 0-59
   - SS is the second, 0-59

   Note: While doing the configuration of this field from tp_shell, the user needs to provide the duration (HHMMSS) in the fields fafDeltaResetTimeHour, fafDeltaResetTimeMinute and fafDeltaResetTimeSecond.

6.4.3 Create an Advanced Filter List

Use filter lists with the FAF's content condition type to filter certain words or phrases.

To create a filter list:

1. In the left navigation bar, select **Advanced Filters ➤ Lists**.
   The Filter List tab appears.
2. Click **Add New**.
   A new Filter List tab appears.
3. Enter a unique name for the list in the **Name** box (up to 31 characters).
4. Optionally enter a description of the list in the **Description** box.
5. In the **Text** box, enter the word(s) and/or phrase(s) the FAF should detect, each on a separate line.

   Note: If this list will be used on an originator field with the **Accuracy** setting "regular expression", it is important to consider the prefix. In the case of a numeric or short code originator, an international number is prefixed with a plus sign (+), a national number is prefixed with an N, and an unknown number is prefixed with a U. In the case of an alphanumeric originator, the originator is prefixed with an A.
6. Click **Save**.
   The MGR creates the filter list and closes the tab.
7. Activate the list.

Chapter 7

Counters

Topics:

7.1 Introduction

The FAF offers many counters that track advanced filters. To retrieve the current value of a specific counter or a group of counters, execute the following command at a command prompt:

```
tp_walk <counter or group name>
```

7.2 FAF Counters

In the MIB, FAF filters are called "chains" and filter conditions are called "blocks". The available FAF counters are:

Counter	Description
fafChainsCounter	Number of times this action was applied
fafChainsTrue	Number of time this filter returned "true"
fafChainsFalse	Number of times this filter returned "false"
fafChainsRet	Number of times this filter caused an immediate return
fafBlocksTrue	Number of times this filter condition returned "true"
fafBlockFalse	Number of times this filter condition returned "false"
fafBlocksRet	Number of times this filter condition caused an immediate return
fafEciSentConnects	Number of connection requests sent
fafEciRcvdConnects	Number of connection requests received
fafEciGotConnects	Number of successful connections
fafEciSentClose	Number of close requests sent
fafEciRcvdClose	Number of close requests received
fafEciSentBytes	Number of bytes sent
fafEciRcvdBytes	Number of bytes received
fafEciSentMsgs	Number of packages sent
fafEciRcvdMsgs	Number of packages received
fafEciSentPartial	Number of congestions
fafEciRcvdPartial	Number of partial receives
fafEciTimeDisconnected	Duration spent disconnected, in seconds

Counter	Description
fafEciTimeConnecting	Duration spent connecting, in seconds
fafEciTimeConnected	Duration spent connected, in seconds
fafEciTimeCongested	Duration spent in congestion, in seconds (this time is also counted as connected)
fafEciRcvdRequests	Number of received evaluation requests
fafEciSentResponses	Number of sent evaluation responses
fafEciReqdNotifications	Number of requested notification indications
fafEciRcvdNotifications	Number of received notification indications
fafEciMissNotifications	Number of missed notification indications
fafEciSentLogin	Number of sent log-in requests
fafEciRcvdLogin	Number of received log-in confirmations
fafEciRefusedLogin	Number of received negative log-in responses
fafEciGrantedLogin	Number of received positive log-in responses
fafEciSentLifecheck	Number of sent life checks
fafEciRcvdLifecheck	Number of received life check confirmations
fafEciSentLogout	Number of sent logout requests
fafEciRcvdLogout	Number of received logout confirmations
fafEciLostSync	Number of disconnections due to lost synchronization
fafEciSentConfirmations	Number of sent notification confirmations
fafEciRcvdErrors	Number of received error messages
fafEciUnknownTags	Number of unknown ASN.1 tags received
fafEvalCountTestExpressionErrors	Number of malformed test expressions evaluated
fafEvalCountAssignmentExpressionErrors	Number of malformed assignment expressions evaluated
fafStringMatches	Total number of matched strings
fafDupsCountMsgs	Total number of messages that have matched this cluster

Counter	Description
fafBulkCountMessages	Amount of messages that have been filtered out for this particular variable value
fafBlackListCntTotalProvisioningRequests	Total number of blacklist provisioning requests from the FAF to the SPF.
fafBlackListCntSuccessfulProvisioningRequests	Total number of successfully processed blacklist provisioning requests sent from the FAF to the SPF.
fafBlackListCntFailedProvisioningRequests	Total number of failed blacklist provisioning requests sent from the FAF to the SPF.
fafBlackListGtNetCntTotalProvisioningRequests	Total number of blacklist provisioning requests sent from the FAF to the MGR.
fafBlackListGtNetCntSuccessfulProvisioningRequests	Total number of successfully processed provisioning requests sent from FAF to MGR.
fafBlackListGtNetCntTimeDroppedProvisioningRequests	Total number of dropped provisioning requests sent from FAF to MGR due to time expiry.
fafBlackListGtNetCntRetryDroppedProvisioningRequests	Total number of dropped provisioning requests sent from FAF to MGR due to retry complete.

Chapter 8

License

Topics:

8.1 Introduction

All FAF filters are enabled or disabled by entering the appropriate license key.

Note: Installing the FAF always requires the installation of a valid license file; otherwise the software will not function.

8.2 Licensed Items

The following table lists the conditions that are controlled by the product license.

Licensed Item	Possible Values
String Module (Content)	Enabled / Disabled
Duplicates Module	Enabled / Disabled
Flooding Module	Enabled / Disabled
EMS Module	Enabled / Disabled
Bulk Module	Enabled / Disabled
Expression Module	Enabled / Disabled
Volume Module	Enabled / Disabled
Auto Blacklist	Enabled / Disabled
Auto GT Network Blacklist	Enabled / Disabled

8.2.1 Multi-Instance License

Multi-instance feature allows you to run multiple FAFs (up to 10 instances) on the same node. Multi-instance license should be enabled for ZMM user to run one additional instance of FAF. To run one additional instance of FAF from newly created ZMM user (using script `tp_manage_user`), an instance license for newly create user id (operating system user identifier) should be enabled in the license file.

8.3 Checking the License Settings

To view the current license values, execute the following command at the command prompt:

```
tp_system --tp_faf [system]
```

Where `[system]` is the IP address or host name of the FAF. If `[system]` is omitted, the local host will be examined. The command must be executed while the FAF is running on the specified node.

8.3.1 Sample tp_system Output

The following sample is a typical example of the output of the `tp_system` command.

```
Identification:
    TextPass/FAF R02.11.04.00
    Linux 3.10.0-862.14.4.el7.x86_64  #1 SMP Fri Sep 21 09:07:21 UTC 2018
    Linux build

Uptime:
    6 days 18h:33m:12s

License key:
   XXXX-XXXX-XXXX-XXXX-XXXX-XXXX-XXXX-XXXX-XXXX

License information:
    License version 8
    License number TST-93782
    Hardware ID 287567c3
    SS7 Board No
    Ext Serial No SGH509XRXJ
    Instance Serial No 200
    License exceeds in 1812 hours
    License exceeds at Tue Feb 19 04:30:00 2019
    License issue number 16
    Hide User Data in Events disabled
    VmWare Support disabled
    Core Count 0
    Encrypt User Data disabled
    String     Module enabled
    Duplicates Module enabled
    Flooding   Module enabled
    EMS        Module enabled
    Bulk       Module enabled
    Expression Module enabled
    Volume     Module enabled
    Auto Blacklist     Module enabled

Alarm stations:
    127.0.0.1:11173
```

If multi-Instance license is enabled for user id 200 (`textpass`) in license file then `tp_system` will display the instance user id as shown below:

```
Instance Serial No 200
```

8.4 Activating a New License

A new license key is required to activate new services or adapt connectivity or performance settings. Please contact your ZephyrTel account manager to obtain a new license key.

To activate a new license key:

1. Place a valid FAF license file in the `TextPass/etc` directory.
2. At the command prompt, execute the following command:

   ```
   tp_system --tp_faf --read_licensekey [system]
   ```

   Where [system] is the IP address or host name of the FAF.

Alternatively, activate the new license by restarting the FAF (refer to *System Management*).

To verify the new license, execute the following command:

```
tp_system --tp_faf --show_licensekey [system]
```

Where `[system]` is the IP address or host name of the FAF.

8.5 License Warnings

The following SNMP traps warn when the license limits are approaching:

- `licenseWillExpire`—The temporary license will expire in the given number of hours.
- `licenseExpired`—The validity period of the license has expired.

The traps are generated at these intervals:

- `licenseWillExpire`—Before it expires, 14 days in advance
- `licenseWillExpire`—7 days in advance
- `licenseWillExpire`—3 days in advance
- `licenseWillExpire`—1 day in advance
- `licenseExpired`—At the moment it expires and then every hour until fixed

Note: To adapt the license in a timely manner and avoid expiry, these traps should be properly handled by the Network Management System.

Chapter 9

System Management

Topics:

9.1 Introduction

This chapter describes the command-line tools that are available to assist in controlling the FAF, determining its status, and locating the causes of issues.

9.2 Stopping the System

The FAF is designed to run unattended and almost maintenance-free for normal operation. If the FAF process needs to be stopped, execute the following command at the command prompt of the FAF node:

```
tp_stop --tp_faf
```

This command will gracefully shut down the FAF system and stop the FAF process and the watchdog process.

9.3 Starting the System

To start the FAF process, execute the following command at the command prompt of the FAF node:

```
tp_start --tp_faf
```

This command will start the FAF and the watchdog process.

During initialisation, the FAF process uses the `tp_config` tool to load the configuration.

When the FAF node restarts after an unplanned outage (such as a power failure), the FAF process is restarted automatically and resumes service.

9.4 Watchdog Process

The watchdog process and the Mobile Messaging component process communicate via Unix signals.

The watchdog process expects contact from the Mobile Messaging component process every second. If the component process does not contact the watchdog for six seconds, the watchdog stops and restarts the component process.

If a signal is missed, the watchdog writes the following message in the syslog:

```
Missing health signal, missed <number of signals> signals, allowed <max number of
missed signals>
```

If the watchdog stops the component process, it writes the following messages:

```
Missed <number of signals> health signals: trying to cleanly abort process <process
 ID>
Application killed (<process ID>), waiting <number of seconds> seconds before
restarting
```

When the watchdog attempts to restart the component process, it writes the following message:

```
Application restarted, number of unsuccessfully restarts <number of restarts>,
application was running for <number of seconds> seconds
```

If the component process dies or is stopped by the watchdog three times within 30 minutes, the watchdog stops attempting to restart the process and writes the following message:

```
Application terminated, too many restarts within predefined interval
```

To monitor the `syslog`, execute:

```
# tail -f /var/log/messages
```

9.5 System Verification

9.5.1 Basic System Verification

To verify the status of FAF, log in as user textpass and execute:

```
$ tp_system --tp_faf [system]
```

Where `[system]` is the resolvable host name or the IP address of FAF.

The response contains the amount of time that the FAF process has been running. If there is no response, the specified system cannot be reached or FAF is not running correctly.

When you use `tp_start`, the FAF process starts to run in the background. For debugging purposes, you can start the FAF process in the foreground:

1. Open a terminal session on the FAF server, log in as user textpass, and start all Mobile Messaging components:
   ```
   $ tp_start
   ```
2. Stop only the FAF process:
   ```
   $ tp_stop --tp_faf
   ```
3. After the FAF process stops, execute:
   ```
   $ usr/TextPass/bin/tp_faf --fg --stderr --debug=0xfff
   ```
4. Open a second terminal session on the FAF server and log in as user textpass, then execute:
   ```
   $ /usr/TextPass/bin/tp_config --tp_faf
   ```

The first terminal session will display the processes being executed and any errors that occur.

9.5.2 Advanced System Verification

FAF configuration parameters and counters provide a more detailed view of the FAF status. To retrieve this information, execute the following command:

```
tp_walk --tp_faf <object>
```

Where `<object>` uniquely identifies an SNMP object, group, or table of interest on the FAF. Relevant objects are documented in the MIB. The following table provides a brief overview.

Object	Description
fafBlocksTable	Shows all filter conditions
fafBulkCountTable	Shows up to 500 records currently marked as bulk
fafBulkTable	Shows all bulk filter instances
fafChainsTable	Shows all filters
fafDupsCountTable	Shows extra information for duplicate filters
fafDupsTable	Shows all duplicate filter instances
fafEciTable	Shows all ECI connections
fafEmsTable	Shows all EMS filter instances
fafFloodCountTable	Shows extra information for flooding filters
fafFloodTable	Shows all flooding filter instances
fafListTable	Shows all lists
fafStringTable	Shows all content filter instances

9.5.2.1 FAF ECI Operational States

You can view the ECI operational state of each installed FAF by executing the following command at a command prompt:

```
tp_walk --tp_faf fafEciTable
```

The MIB property `fafEciOperationalState.n` (where `n` is the index of the FAF in the MGR) shows the FAF's ECI operational state.

The possible operational states are:

- 1—Disconnected
- 2—Connecting
- 3—Connected
- 4—Reconnecting

For example:

```
fafEciOperationalState.1 = INTEGER: connected(3)
fafEciOperationalState.1 = INTEGER: disconnected(1)
```

9.5.2.2 Additional Debugging Methods

Additional methods to debug the FAF are:

- Examine a core file
- Analysis of XML configuration files

9.6 Command-Line Tools for Troubleshooting

The following commands are available to troubleshoot the FAF.

```
tp_system --tp_faf --traps
```

Makes `tp_system` behave as a simple SNMP trap receiver, monitoring the "memory exhausted" and "device operational state changed" traps.

```
tp_walk --tp_faf mbalance
```

Performs an SNMP walk of the entire MIB of the FAF.

9.7 Commands for Troubleshooting

The following operating system tools are available for troubleshooting:

Tool	Description
`gcore`	Generates a core file
`hostid`	Provides the host ID of the system (required for a license)
`ifconfig`	Provides an overview of the IP configuration
`netstat`	Provides an overview of IP network statistics
`ping`	An IP diagnostic command
`top`	Shows statistics about active processes
`ps`	Provides information about processes
`sar`	Provides performance data
`tcpdump`	Trace network traffic on TCP/IP level

Refer to the Red Hat Enterprise Linux documentation at *http://www.redhat.com/docs/*.

Appendix A

ECM and ABL Configuration Example

Topics:

A.1 Introduction

This appendix provides an example of the configuration required on the Manager (MGR) to configure External Condition Message (ECM) functionality. This is an example for a virus alert SMS.

Apart from detecting the SMS flooding from an external originator and sending back an appropriate virus alert message to that subscriber, all subsequent messages from the same originator would also be blocked for the next 1 day by means of the Automatic Blacklisting functionality.

Note: The MGR information in this section is an example only. However, Auto Blacklist license must be enabled.

A.2 Advanced Filters > Filters

Field	Sample
Name	FloodingDetection
Description	Setting EC Message when flooding
Priority	50
Action	Return False
Blacklist Party	Originator
Blacklist Service	Orig_srv_abl
Blacklist Action	Time-based Blocking
Blacklist Duration	DD: 01 HH: 00 MM:00

A.3 Advanced Filters > Filter Conditions > FloodingDetection

Field	Sample
Filter Name	FloodingDetection
Priority	40
Description	Flooding
Invert	Checked
Type	Flooding
Flood Condition	Flooding

A.4 Advanced Filters > Filter Conditions > FloodingDetection

Field	Sample
Filter Name	FloodingDetection
Priority	50
Description	Expression
Invert	Unchecked
Type	Expression
Flood Condition	text expression

A.5 Advanced Filters > Conditions > Flooding

Field	Sample
Name	Flooding
Description	Flooding
Field	Orig
Significant Digits	8
Minimal Traffic	10
Traffic Increase Rate	10
Time Delay	10
Filter Period Flooding	1
Filter Period Baseline Traffic	3600
Margin	5

A.6 Advanced Filters > Conditions > Expression Condition

Field	Sample
Name	text expression
Description	text expression
Text Expression	
Assignment Expression	failuremessagekey = 1

A.7 EC Applications > Messages

Field	Sample
Message Name	Virus alert SMS
Match on	Specific Field and Code
Condition Field	Message Key
Result Code	1
Specific EC Application	FAF
External Condition Result	False Only (or Any)
Enabled for	MO Message
Message Template	Your message to $(DESTINATION) was not sent, because your handset might be infected by a virus. Please report code $(CODE) to our Customer Care Center.
Message Originator	<Operator Name>
Message Recipient	Original Originator
Suppress Status Report	Checked

Appendix B

Sample Configuration File

Topics:

B.1 Common Configuration File

Using this configuration file, the FAF will attempt to set up two ECI connections and two ISPI connections. SNMP traps are sent to UDP port 11173 on the local host. `tp_config` executes the post-boot scripts after the static node parameters have been provisioned.

```
<tpconfig
    ipaddress="127.0.0.1"
    runtpfclientprocess="true"
    runtextpassprocess="false"
    runtextfafprocess="true"
    >

<!--
    The default value of wordboundary is the following.
    wordboundary=" ~`!@#$%^&amp;*()+-_=[]{}\\|:&quot;;'&lt;&gt;?,./"
    You can add the wordboundary to fafprop to customize word boundary.
-->
    <fafprop
        normalisationmap="0oO&#246;&#214;&#10;1iIlL!\/&#10;2zZ&#10;3eE&#10;
        4aA&#228;&#196;&#10;5sS&#223;&#10;6&#10;7tT&#10;8bB&#10;9gG&#10;
        cC&#10;dD&#10;fF&#10;hH&#10;jJ&#10;kK&#10;mM&#10;nN&#10;pP&#10;
        qQ&#10;rR&#10;uU&#252;&#220;&#10;vV&#10;wW&#10;xX&#10;yY"

enableispiconntcpkeepalive=”true”
     />

    <fafeci host="10.0.0.13" port="9500" user="user1" pass="pass1"/>
    <fafeci host="127.0.0.1" user="user2" pass="pass2"/>

<!-- SPF configuration -->
    <fafspf name="Core-1" ipaddress= "10.0.0.119" port="9800"/>
    <fafspf name="Core-2" ipaddress= "10.0.0.120” port="9800"/>

    <trapreceiver ipaddress="127.0.0.1" udpport="11173"/>
</tpconfig>
```

B.2 Host-Specific Configuration File

```
<tpconfig
    runtpfclientprocess="true"
    runtpfclientprocess="false"
    runtextpassprocess="false"
    runtextfafprocess="true"
    ipaddress="127.0.0.1"
    >

    <trapreceiver ipaddress="127.0.0.1" udpport="11173"/>
</tpconfig>
```

Appendix C

References

Topics:

C.1 References

1. 3rd Generation Partnership Project (3GPP) Technical Specification 23.040 Release 9 at *http://www.3gpp.org*
2. UTF-8 (RFC 2279) at *http://www.ietf.org/rfc/rfc2279.txt*
3. IEEE POSIX Extended Regular Expression standard (1003.2-1992) at *http://www.opengroup.org/onlinepubs/009695399/basedefs/xbd_chap09.html#tag_09_04*
4. Unicode at *http://www.utf8-chartable.de/*
5. ZephyrTel Mobile Messaging MGR Operator Manual
6. ZephyrTel Mobile Messaging SPF Operator Manual
7. ZephyrTel Mobile Messaging RTR Operator Manual
8. ZephyrTel Mobile Messaging SNMP Trap Reference Guide
9. ZephyrTel Mobile Messaging Full Element Installation Manual

Glossary

#

3GPP	3rd Generation Partnership Project

A

ABL	Automatic Blacklisting An enhanced anti-spam and anti-fraud functionality, wherein the FAF filters screen incoming MO/MT messages received from the RTR and, if a message is detected as `spam` or `fraudulent` based on the appropriately configured filter conditions, sends an automatic provisioning request to the SPF to blacklist the corresponding originator or recipient subscriber for either a specified duration of time or permanently.
AO	Application Originated Short message traffic that is originated by an application.
application	The telecommunications software that is hosted on the platform. A service provided to subscribers to a network; for example, voice over IP (VoIP), video on demand (VoD), video conferencing, or gaming.
ASCII	American Standard Code for Information Interchange
AT	Application Terminated Short message traffic that terminates at an application.

C

C

CPU	Central Processing Unit

D

DCS	Data Coding Scheme

E

EC	External Condition Condition that is passed on the external condition interface.
ECI	External condition interface Interface for communicating with external condition applications.
ECM	External condition message Message that is passed on the external condition interface.
EMS	Element Management System The EMS feature consolidates real-time element management at a single point in the signaling network to reduce ongoing operational expenses and network downtime and provide a higher quality of customer service.

F

FAF	Firewall Advanced Filter Works in combination with the Firewall to filter messages, modify message content, and alert network operators of increases in SMS-related traffic.
FWL	Firewall Helps protect subscribers from receiving unwanted messages and provides statistical information and

F

	message details about inbound suspect messages.

G

GT	Global Title Routing Indicator
GUI	Graphical User Interface The term given to that set of items and facilities which provide the user with a graphic means for manipulating screen data rather than being limited to character based commands.

I

IMSI	International Mobile Subscriber Identity
IP	Internet Protocol IP specifies the format of packets, also called datagrams, and the addressing scheme. The network layer for the TCP/IP protocol suite widely used on Ethernet networks, defined in STD 5, RFC 791. IP is a connectionless, best-effort packet switching protocol. It provides packet routing, fragmentation and re-assembly through the data link layer.

M

MB	Megabyte — A unit of computer information storage capacity equal to 1,048, 576 bytes.
MGR	A Web-based interface for managing ZephyrTel Mobile Messaging components. Prior to Suite 6, the Configuration Manager (CM) provided this functionality.

M

MIB	Management Information Database
MO	Mobile Originated Refers to a connection established by a mobile communication subscriber. Everything initiated by the mobile station is known as mobile originated.
MSC	Mobile Switching Center
MSISDN	Mobile Station International Subscriber Directory Number The MSISDN is the network specific subscriber number of a mobile communications subscriber. This is normally the phone number that is used to reach the subscriber.
MT	Mobile Terminated All transmissions that reach the mobile station and are accepted by it, such as calls or short messages.
MTIX	Incoming mobile-terminated external condition External condition (EC) rule that operates on incoming mobile-terminated (MT) messages.
MTOX	Outgoing mobile-terminated external condition External condition (EC) rule that operates on outgoing mobile-terminated (MT) messages.

N

NPS	Non-Provisionable Service

N

A service that cannot be provisioned by the subscriber. For example, the subscriber is not able to switch the service ON/OFF or provision the service with service specific settings.

P

ping

A network tool used to determine if a target host can be reached across an IP network. Ping estimates the round-trip time and packet loss (if any) rate between hosts.

R

RFC

Request for Comment

RFCs are standards-track documents, which are official specifications of the Internet protocol suite defined by the Internet Engineering Task Force (IETF) and its steering group the IESG.

RTR

Router

Routes all types of SMS traffic.

S

SCCP

Signaling Connection Control Part

SM

Short Message

SMPP

Short Message Peer-to-Peer Protocol

An open, industry standard protocol that provides a flexible data communications interface for transfer of short message data.

SMS

Short Message Service

S

SMSC	Short Message Service Center
SNMP	Simple Network Management Protocol. An industry-wide standard protocol used for network management. The SNMP agent maintains data variables that represent aspects of the network. These variables are called managed objects and are stored in a management information base (MIB). The SNMP protocol arranges managed objects into groups.
SS7	Signaling System #7
STP	Signal Transfer Point The STP is a special high-speed switch for signaling messages in SS7 networks. The STP routes core INAP communication between the Service Switching Point (SSP) and the Service Control Point (SCP) over the network.

T

TCP	Transfer Control Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
trap	A mechanism used in the context of SNMP (Simple Network Management Protocol) for one-way event notification.

U

UDH	User Data Header

U

UDP	User Datagram Protocol
UTF-8	Variable-length character encoding for Unicode that is backward-compatible with ASCII.

X

XML	eXtensible Markup Language A version of the Standard Generalized Markup Language (SGML) that allows Web developers to create customized tags for additional functionality.