



# FS – Traffic Event Sender

## Content

<b>1</b>	<b>INTRODUCTION .....</b>	<b>2</b>
<b>2</b>	<b>FUNCTION REQUIREMENTS (COMMERCIAL).....</b>	<b>2</b>
<b>3</b>	<b>FUNCTION SPECIFICATION (DESIGN RELATED) .....</b>	<b>2</b>
3.1	INTRODUCTION .....	2
3.2	EXPORTED INTERFACES .....	3
3.2.1	<i>ITrafficEvent</i> .....	3
3.2.2	<i>ITrafficEventSender</i> .....	3
3.2.3	<i>IEventReceiver</i> .....	3
3.3	IMPORTED INTERFACES .....	4
3.4	FUNCTIONS .....	4
3.4.1	<i>Report Traffic Events</i> .....	4
3.4.2	<i>Available Properties</i> .....	7
3.4.3	<i>Events</i> .....	11
3.4.4	<i>Thread Safe</i> .....	11
3.4.5	<i>Abstract layers and encapsulation</i> .....	11
<b>4</b>	<b>EXTERNAL OPERATION CONDITIONS.....</b>	<b>11</b>
4.1	CONFIGURATION .....	11
4.1.1	<i>General Configuration for RADIUS-MA Usage</i> .....	12
4.1.2	<i>General Configuration for Email Usage</i> .....	12
4.1.3	<i>Configuration per Traffic Event Type</i> .....	12
4.2	LOGGING .....	13
<b>5</b>	<b>CAPABILITIES.....</b>	<b>13</b>
<b>6</b>	<b>REFERENCES .....</b>	<b>13</b>
<b>7</b>	<b>TERMINOLOGY .....</b>	<b>14</b>

## History

Version	Date	Adjustments
A	2006-10-03	First version. (MANDE)
B	2007-06-13	Minor changes. Updated with new attributes (ermkese)
C	2008-02-25	Added ESS Information.



D	2008-08-05	Merge from Mobinil to DP5. Updated with a new attribute. (emahagl)
---	------------	--

## 1 Introduction

This document specifies the function of the Traffic Event Sender component. The Traffic Event Sender can be used by clients to notify that a Traffic Event has occurred. The Traffic Event Sender relays that information to M3 components that are interested in this type of information, e.g. MER or NTF.

## 2 Function Requirements (Commercial)

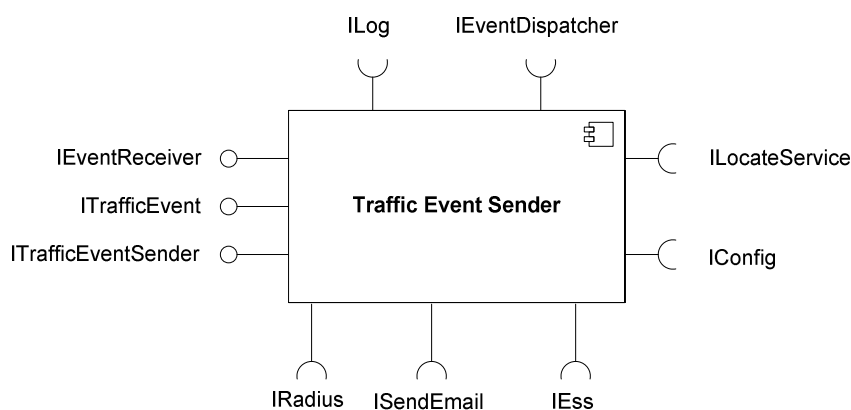
This paragraph is intentionally left blank.

## 3 Function Specification (Design Related)

### 3.1 Introduction

Using the Traffic Event Sender, clients can notify M3 components of Traffic Events that have occurred.

The purpose of the Traffic Event Sender is only to relay Traffic Event information to M3 components. Traffic Event information is sent either as an email or using the RADIUS-MA protocol (see [2]).



**Figure 1 Traffic Event Sender component**



## 3.2 Exported Interfaces

### 3.2.1 ITrafficEvent

This interface is used to create a Traffic Event. Each Traffic Event is uniquely identified by its name. The event name is also used to identify the event configuration.

#### 3.2.1.1 Methods

ITrafficEvent offers the following methods:

- *String getName()*  
Returns the name defining the type of the Traffic Event.
- *Object getProperty(String propertyName)*  
Returns the specified Traffic Event property.
- *void setName(String trafficEventName)*  
Sets the name of the Traffic Event.
- *void setProperty(String propertyName, Object propertyValue)*  
Sets the specified Traffic Event property to the given value.

### 3.2.2 ITrafficEventSender

This interface is used to report that a Traffic Event has occurred.

#### 3.2.2.1 Methods

ITrafficEventSender offers the following methods:

- *void reportTrafficEvent(ITrafficEvent trafficEvent)*  
Sends the Traffic Event to interested parties. How to distribute the Traffic Event is configurable (see 4.1.4.1).
- *void reportTrafficEvent(ITrafficEvent trafficEvent, Restrictions restrictions)*  
Sends the Traffic Event to interested parties. How to distribute the Traffic Event is configurable (see 4.1.4.1). The Restriction parameter is used to list receiving parties that should not receive this Traffic Event, see 3.2.2.2.

#### 3.2.2.2 Parameter Type Description

##### 3.2.2.2.1 Restrictions

The Restrictions parameter is used to control which type of receivers that are not allowed to receive this Traffic Event. With this parameter, the client reporting the Traffic Event can inform Traffic Event Sender that end-users are not allowed to be notified of this Traffic Event. This is used by the Traffic Event Sender to decide on event distribution restrictions, e.g. to avoid sending a specific event as an email.

### 3.2.3 IEventReceiver

This interface is used to receive events generated by other components. For details in the interface, see [4].



## 3.3 Imported Interfaces

The Traffic Event Sender uses the following external interfaces:

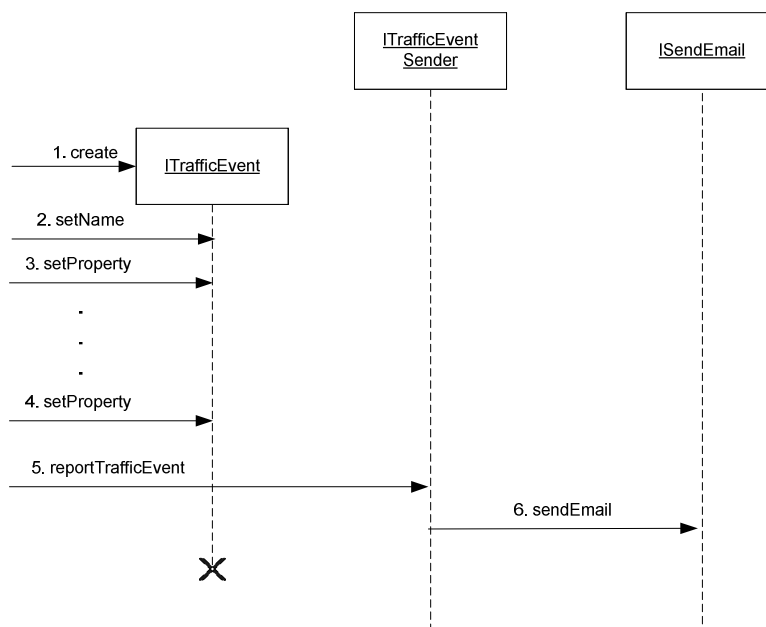
- *ILog* for logging.
- *IConfig* for configuration.
- *ILocateService* for M3 service lookup.
- *ISendEmail* to send the Traffic Event as an email.
- *IRadius* to send the Traffic Event over RADIUS-MA.
- *IEss* to send TrafficEvent/Message to Event Subscription Server (ESS).
- *IComponentManager* to retrieve an instance of another component.
- *IEventDispatcher* for receiving events, see [4].

## 3.4 Functions

### 3.4.1 Report Traffic Events

When reporting a Traffic Event, Traffic Event Sender sends the Traffic Event to interested parties using email, RADIUS-MA, ESS or both email and RADIUS-MA.

#### 3.4.1.1 Using Email



**Figure 2 Reporting a Traffic Event using email**

Figure 2 exemplifies reporting a Traffic Event that is sent using email:

1. A client creates a new Traffic Event.
2. The name of the Traffic Event is set by the client.
3. Properties of the Traffic Event are set by the client.



4. This is done until all properties are set.
5. When all the properties have been set, the Traffic Event is reported to the Traffic Event Sender.
6. The Traffic Event is sent as an email.

The Traffic Event Sender uses the external interface `ISendEmail` (see [6]) in order to send the Traffic Event as an email.

#### 3.4.1.1.1 Email Host

When the Traffic Event is sent as an email, it is either sent to the preferred host or to a specified host. This depends on the receiver of the event email. If all receivers belong to the same mail host, this specified host is used; otherwise a preferred host is used. If the preferred host is used, the Traffic Event Sender simply lets `ISendEmail` select a suitable host, see [6]. If a specified host is used, this is submitted to `ISendEmail`.

If the sending of the email is unsuccessful, this is logged as an error. No retries are performed by the Traffic Event Sender since they have already been performed by the Message Sender.

#### 3.4.1.1.2 Buffering of Traffic Events

The Traffic Event Sender can be configured to buffer Traffic Events of the same type before sending them together in one email. The Traffic Events are buffered until a certain amount of messages have been buffered or a certain amount of time has elapsed since the last email was sent. The message limit and time limit are configurable both generally (see 4.1.2) and per Traffic Event type (see 4.1.4.1.1).

#### 3.4.1.2 Using RADIUS-MA

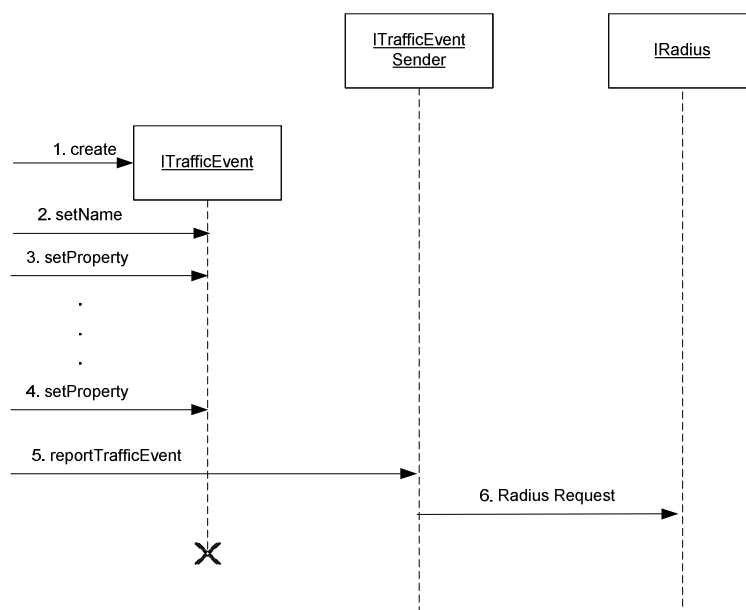


Figure 3 Reporting a Traffic Event using RADIUS-MA



Figure 3 exemplifies reporting a Traffic Event that is sent using RADIUS-MA:

1. A client creates a new Traffic Event.
2. The name of the Traffic Event is set by the client.
3. Properties of the Traffic Event are set by the client.
4. This is done until all properties are set.
5. When all the properties have been set, the Traffic Event is reported to the Traffic Event Sender.
6. The Traffic Event is sent using RADIUS-MA.

#### 3.4.1.2.1 Radius Host

The Traffic Event Sender uses the external interface `ILocateService` (see [5]) to retrieve MER service instance. In case of a MER failure, this is logged as an error and the MER host is reported using the `ILocateService` interface.

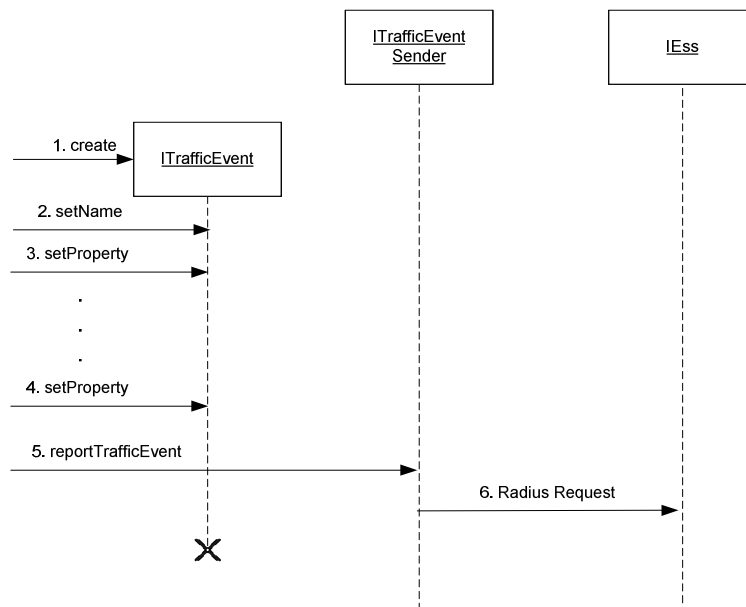
#### 3.4.1.2.2 Queuing Traffic Events

When a Traffic Event shall be sent using RADIUS-MA, the Traffic Event Sender uses a "fire-and-forget" mechanism, i.e. the Traffic Event will be put on a queue and the `reportTrafficEvent` method will return immediately.

The Traffic Events are then popped from the queue in the order they arrived and sent over RADIUS-MA to MER. In case of a RADIUS failure, the event is resent a configurable amount of times, before another MER is tried. This is logged as an error and the failing MER is reported using the `ILocateService` interface. In case of a MER failure, the Traffic Events are kept in the queue. The RADIUS counter for the failed event is retained. The queue size is configurable (see 4.1.1) and when the queue limit is reached, the received Traffic Event will be dropped.

- The queue is not persistent, i.e. all Traffic Events are lost if the Traffic Event Sender is restarted.

### 3.4.1.3 Using Event Subscription Server



**Figure 4 Reporting a Traffic Event using Event Subscription Server**

Figure 3 exemplifies reporting a Traffic Event that is sent using Event Subscription Server:

- 7.** A client creates a new Traffic Event.
- 8.** The name of the Traffic Event is set by the client.
- 9.** Properties of the Traffic Event are set by the client.
- 10.** This is done until all properties are set.
- 11.** When all the properties have been set, the Traffic Event is reported to the Traffic Event Sender.
- 12.** The Traffic Event is sent using ESS.

#### 3.4.1.3.1 ESS Host

The Traffic Event Sender uses the external interface `ILocateService` (see [5]) to retrieve ESS service instance. In case of a ESS failure, this is logged as an error and the ESS host is reported using the `ILocateService` interface.

#### 3.4.1.3.2 Queuing Traffic Events

When a Traffic Event shall be sent using ESS API, the Traffic Event Sender uses a "fire-and-forget" mechanism, i.e. the Traffic Event will be put on a queue in the ESS API and the `reportTrafficEvent` method will return immediately.

The Traffic Events are then popped from the queue in the order they arrived and sent over ESS API to Event Subscription Server. In case of an ESS failure, the ESS API reports to Traffic event sender that the ESS host is down. This is logged



as an error and the failing ESS host is reported using the ILocateService interface. the resend mechanism is handled by the ESS API, see[].

### 3.4.2 Available Properties

The available Traffic Event properties vary depending on if the Traffic Event shall be sent using email or RADIUS-MA.

The event properties are defined in the configuration, see 4.1.1 and 4.1.4.2. Table 1 lists the initially required properties which will be supported and suggests in which situations they would be used; email, RADIUS-MA or both.

**Table 1 Initially required properties**

Property Name	Type	Used in email	Used in RADIUS-MA	Description
sessionid	String		X	Uniquely identifies a session in the system.
username	String	X	X	Identifies a user, e.g. a phone number or an email address
callednumber	String	X	X	The called phone number involved in the Traffic Event.
callingnumber	String	X	X	The calling phone number involved in the Traffic Event.
product	Integer		X	Specifies the client product.
operation	Integer		X	The type of operation used in the Traffic Event.
messagesize	Integer		X	The size of a message or part of a message. Only of interest if the Traffic Event regards a message.
terminatecause	Integer		X	The reason for a Traffic Event indicating failure. Only of interest if a Traffic Event indicates a failure.
accesstype	Integer	X	X	The type of access into the M3 system.
objecttype	Integer		X	The type of object involved in the Traffic Event, e.g. a user or a message.
objectid	String		X	The identifier of the object involved in the Traffic Event.
description	String		X	A description of the Traffic Event.
messagetype	Integer		X	Type of message. Only of interest if the Traffic Event regards a message.





Approved: Per Berggren

No: 11/FS-MAS0001 Uen

Copyright Mobeon AB  
All rights reservedAuthor: Kenneth Selin  
Title: FS - Traffic Event SenderVersion: D  
Date: 2008-08-05

9/14

Property Name	Type	Used in email	Used in RADIUS-MA	Description
messageencoding	Integer		X	The encoding of the message. Only of interest if the Traffic Event regards a message.
mailhost	String	X		Subscriber's mail host, see [3].
sender	String	X		The phone number of the sender of the message, see [3].
destination	String	X		The phone number of a recipient, see [3].
language	String	X		The language of the receiver of the message, see [3].
date	Date	X		A date and time regarding the Traffic Event, see [3].
action	String	X		An action regarding the Traffic Event, see [3].
forwardingnumber	String	X		The number that should be used to forward calls to the voice mail system, see [3].
unsetforwards	String	X		The forwards that were turned off, see [3].
emailaddress	String	X		The receivers email address, see [3].
mediacontent	Media Object	X		A media content, e.g. a voice or video message.
additionalstatistics information	String		X	Statistical information, e.g. for CAT.
ownername	String		X	See ref. [2].
sssporttype	Integer		X	See ref. [2].
accountcurrency	String		X	See ref. [2].
accountmoney	String		X	See ref. [2].
accounttype	Integer		X	See ref. [2].
accountreason	String		X	See ref. [2].
menuusage	String		X	See ref. [2].



### 3.4.2.1 Mapping to RADIUS-MA

Table 2 illustrates how the initially required properties are mapped to attributes in RADIUS-MA. The value ranges for the properties are defined in [2]. The indicated default value is used when the property has not been set in the Traffic Event. It is considered a failure if mandatory properties are not set by entities using the Traffic Event Sender.

**Table 2 Mapping of required properties to RADIUS-MA attributes**

Property Name	Mapped to RADIUS-MA attribute	Required M=Mandatory, O=Optional, C=Conditional
sessionid	Acct-Multi-Session-Id	O
username	User-Name	O
callednumber	Called-Station-Id	O
callingnumber	Calling-Station-Id	O
product	Product-Tag	O If not set, the default value 4 (MoIP) will be used.
operation	Event-Type	M
messagesize	Message-Size	C Mandatory if ObjectType is set to Message.
terminatecause	Terminate-Cause	O
accesstype	SAS-Port-Type	O If not set, the default value 7 (TUI) will be used.
objecttype	Object-Type	O
objectid	Object-Id	O
description	Event-Description	M
messagetype	Message-Type	C Mandatory of ObjectType is set to Message.
messageencoding	Message-Encoding	C Mandatory of ObjectType is set to Message.
additionalstatistics information	Owner-Name	O
ownername	MAE-Owner-Name	O
sssporttype	SSS-Port-Type	O
accountcurrency	MAE-Account-Currency	O



accountmoney	MAE-Account-Money	O
accounttype	MAE-Account-Type	O
accountreason	MAE-Account-Reason	O
menuusage	MAE-Menu-Usage	O

Apart from the initially required properties, it is possible to add additional properties using the configuration, see 4.1.4.

### 3.4.2.2 Mapping to Email

How Traffic Event properties are mapped to email content is configurable, see 4.1.4.2. This configuration varies between different types of Traffic Events. The email content when notifying NTF of a Traffic Event is described in [3].

In the configuration both the required properties as well as optionally added properties are mapped to email content, e.g. a specific email header.

It is also possible to map Media Objects (see [7]) to an email. It will be added to the email as a MIME part with the Media Object's corresponding content type as Content-type header. Mapping of Media Objects to RADIUS events is not supported.

## 3.4.3 Events

### 3.4.3.1 Consumed

The Traffic Event Sender registers itself (using IEventDispatcher, see [4]) as a receiver of the following events:

- Configuration has changed (indicates that configuration should be re-read)

### 3.4.3.2 Produced

None

## 3.4.4 Thread Safe

The Traffic Event Sender is thread safe.

## 3.4.5 Abstract layers and encapsulation

The parts of the component handling the RADIUS communication are encapsulated.

# 4 External Operation Conditions

## 4.1 Configuration

The configuration of the Traffic Event Sender is divided into three parts:

- General configuration for the RADIUS-MA usage



- General configuration for the email usage
- General configuration for the ESS usage
- Configuration per Traffic Event type

#### **4.1.1 General Configuration for RADIUS-MA Usage**

The following is configurable for usage of the RADIUS-MA protocol:

- Globally unique instance name
- Vendor information
- Size of the Traffic Event queue
- Mapping of Traffic Event properties to RADIUS-MA
- Service name to use in the ILocateService interface

#### **4.1.2 General Configuration for Email Usage**

The following is configurable for email usage:

- Receivers email address
- Maximum number of Traffic Events in buffered email
- Maximum amount of time to wait before sending a buffered email

#### **4.1.3 General Configuration for ESS Usage**

The following is configurable for ESS usage:

- Size of the ESS Event queue
- Service name to use in the ILocateService interface.

The above attributes except ESS attributes, are also configurable per Traffic Event type. The general values are used as default if not specified for a specific Traffic Event type.

#### **4.1.4 Configuration per Traffic Event Type**

Each Traffic Event that shall be possible to report must be configured. The type of the Traffic Event is given by its name. For each Traffic Event type the following is configured:

- The name (used to identify the Traffic Event type)
- A number of configuration parameters (described below)
- Mapping of Traffic Event properties to email (only if the Traffic Event shall be sent using email)
- If the Traffic Event should be transmitted, i.e. it is possible to suppress sending of certain events.



#### 4.1.4.1 Traffic Event Distribution Type

It is configurable per Traffic Event type if the Traffic Event should be sent using email, RADIUS-MA, ESS Message or both email and RADIUS-MA.

##### 4.1.4.1.1 Email

The following is configurable per Traffic Event type if it shall be sent using email:

- If preferred storage or a specific storage should be used (as described in 3.4.1.1.1)
- Whether or not the Traffic Events should be buffered before sent
- Maximum number of Traffic Events in a buffered email
- Maximum amount of time to wait before sending a buffered email
- Receivers mail address

#### 4.1.4.2 Mapping of Traffic Event Properties

The available Traffic Event properties and how they are mapped to an email, ESS and to the RADIUS-MA protocol are configurable in the Traffic Event Sender. For email and ESS, this mapping is done per Traffic Event type. For the RADIUS-MA protocol, this mapping is done once, see 4.1.1.

How to configure the mapping of properties to email content depends on the receiver of the email. For example, for emails sent to NTF the content of the email is described in [3].

To allow for easier configuration, it shall be possible to use templates for events sent as email.

## 4.2 Logging

- Host errors are logged using the IHostedServiceLogger interface, see [8].
- All calls to interface methods are logged at info level. Method name and parameter values are logged. At return returned value is logged.

## 5 Capabilities

This paragraph is intentionally left blank.

## 6 References

- [1] IWD – Extensible Messaging Protocol  
2/155 19-1/HDB 101 02 Uen
- [2] IWD – RADIUS MA  
6/155 19-CRH 109 089 Uen
- [3] IWD – Requesting NTF Services  
5/155 19-CRH 109 127 Uen
- [4] FS – Execution Engine  
3/FS-MAS0001 Uen



- [5]** FS – External Component Register  
9/FS-MAS0001 Uen
- [6]** FS – Message Sender  
6/FS-MAS0001 Uen
- [7]** FS Media Object  
13/FS-MAS0001 Uen
- [8]** FS Log Manager  
2/FS-MAS0001 Uen

## **7 Terminology**

MER	Message Event Repository
NTF	Notification Component