



FS –Message Sender

Content

| | | |
|----------|--|----------|
| 1 | INTRODUCTION | 1 |
| 2 | FUNCTION REQUIREMENTS (COMMERCIAL)..... | 2 |
| 3 | FUNCTION SPECIFICATION (DESIGN RELATED) | 2 |
| 3.1 | INTRODUCTION | 2 |
| 3.2 | EXPORTED INTERFACES | 2 |
| 3.2.1 | ISendEmail | 2 |
| 3.2.2 | IEventReceiver | 3 |
| 3.3 | IMPORTED INTERFACES | 3 |
| 3.4 | FUNCTIONS | 3 |
| 3.4.1 | Send Email | 3 |
| 3.4.2 | Events | 5 |
| 3.4.3 | Thread safe | 5 |
| 4 | EXTERNAL OPERATION CONDITIONS..... | 5 |
| 4.1 | CONFIGURATION | 5 |
| 4.2 | LOGGING | 6 |
| 5 | CAPABILITIES..... | 6 |
| 6 | REFERENCES | 6 |
| 7 | TERMINOLOGY | 6 |

History

| Version | Date | Adjustments |
|---------|------------|------------------------|
| A | 2006-10-03 | First version. (MANDE) |

1 Introduction

This document specifies the function of the Message Sender component. The Message Sender component provides an interface that can be used by other components in order to send e-mail messages.



2 Function Requirements (Commercial)

This paragraph is intentionally left blank.

3 Function Specification (Design Related)

3.1 Introduction

The Message Sender component provides interfaces used by clients to send MIME formatted e-mail messages.

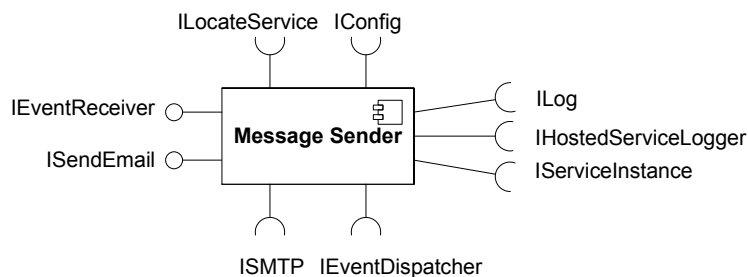


Figure 1 Message Sender component

3.2 Exported Interfaces

3.2.1 ISendEmail

This interface is used to send email messages.

3.2.1.1 Methods

ISendEmail offers the following methods:

- *void sendEmail(Message msg)*
Sends the message as an email to preferred storage.
- *void sendEmail(Message msg, String host)*
Sends the message as an email to the specified host
- *void sendEmail(Message msg, SmtOptions options)*
Sends the message as an email to preferred storage and the SMTP behavior can be change by specifying SMTP options.
- *void sendEmail(Message msg, String host, SmtOptions options)*
Sends the message as an email to the specified host and the SMTP behavior can be change by specifying SMTP options.

3.2.1.2 Parameter Type Description

3.2.1.2.1 Message



This is the email message that shall be sent. Message is given in the JavaMail message format, see [1].

3.2.1.2.2 *SmtOptions*

SmtOptions is a set of key/value pairs to control SMTP the behavior. Supported keys are *envelopeFrom*.

3.2.2 **IEventReceiver**

This interface is used to receive events generated by other components. For details in the interface, see [2].

3.3 **Imported Interfaces**

The Message Sender component uses the following external interfaces:

- *ILog* for logging
- *IHostedServiceLogger* for logging hosted service circumstances.
- *IConfig* for configuration
- *ILocateService* for M3 service lookup
- *IServiceInstance* for accessing properties of looked up M3 services instances
- *IEventDispatcher* to register the component as event receiver.
- *ISMTTP* to send and email

Note that not all interfaces are used during the scenarios and sequences in this document. Configuration, logging, event dispatching and O&M are understood to be used whenever needed and not shown in the scenarios.

3.4 **Functions**

3.4.1 **Send Email**

The Message Sender component can be used to send an email.

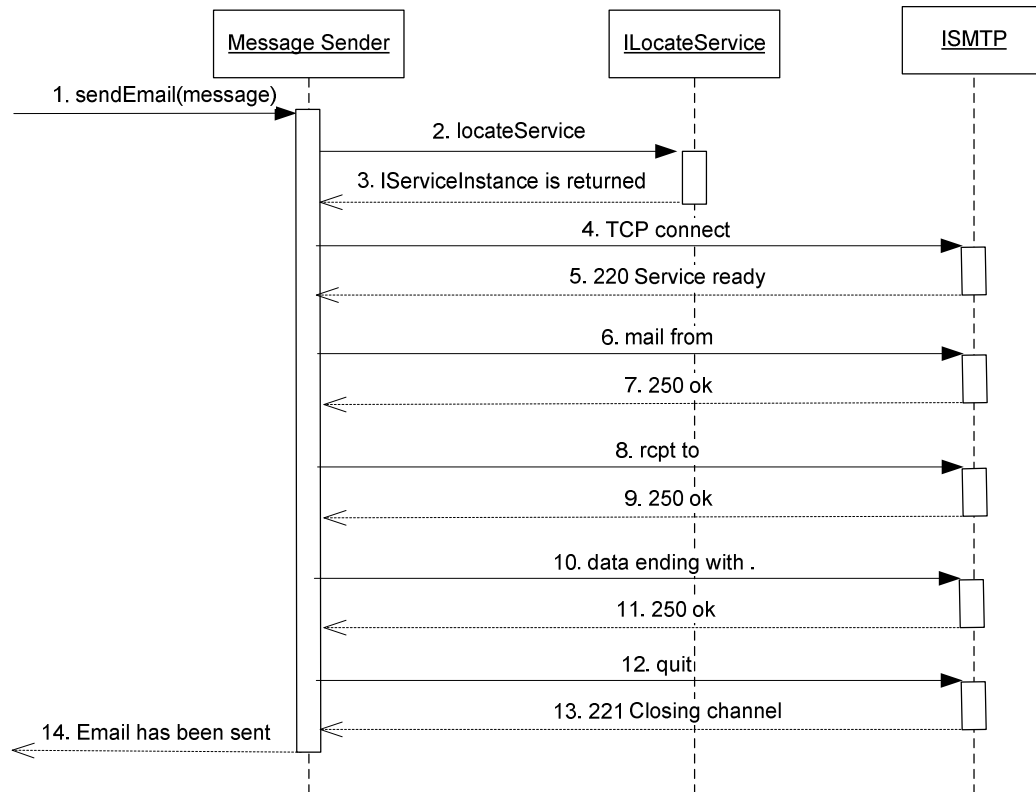


Figure 2 Example of sending an email

Figure 2 exemplifies how an email is sent:

1. A client requests that an email shall be sent.
2. In the example above a host is not specified, therefore the preferred host must be located. This is done using the ILocateService interface.
3. An IServiceInstance is returned.
4. Message Sender connects to the located host.
5. The SMTP host answers indicating that the SMTP service is ready.
6. Message Sender sends the address of the sender.
7. The sender's address is acknowledged with an ok response.
8. Message Sender sends the address of the recipient.
9. The recipient's address is acknowledged with an ok response.
10. Message Sender sends all email data and ends with a single dot.
11. The data is acknowledged with an ok response.
12. The data transfer is completed and the Message Sender sends a quit to end the SMTP session.



13. The connection is closed by the ISTMP server.

14. The sendEmail message returns now that the email has been sent.

If the message cannot be sent (after the configured number of retries) this is logged as an error and notified to the user of the Message Sender.

Sending a message is considered unsuccessful if the sending process not has completed step 11 in the above sequence.

3.4.1.1 JavaMail

The Message Sender component use JavaMail (see [1]) to send the message.

3.4.1.2 SMTP Host and Port

The SMTP host and port is always extracted from an IServiceInstance. The IServiceInstance is located with the ILocateService interface, see [3]. Message Sender delegates to ILocateService to choose a preferred host to be returned.

If the message cannot be sent with the located IServiceInstance, another one should be located for the next try.

If a host was specified in the call to *sendEmail*, that host is provided along with the service name when looking up the IServiceInstance to the first try to send message.

The value of the service name that Message Sender uses when locating the IServiceInstance is configurable.

3.4.2 Events

3.4.2.1 Consumed

The Message Sender component registers itself (using IEventDispatcher, see [2]) as a receiver of the following events:

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

3.4.2.2 Produced

None

3.4.3 Thread safe

The Message Sender component is thread safe.

4 External Operation Conditions

4.1 Configuration

The following properties are configurable in Message Sender:

- Timeout of SMTP.
- Amount of SMTP send tries.



- Service name used to locate the ISMTP service.

4.2 Logging

- Hosted service circumstances should be logged as described in ref [4] using the IHostedServiceLogger interface.
- All calls to interface methods are logged at info level. Method name and parameter values are logged. At return is returned value logged.

5 Capabilities

This paragraph is intentionally left blank.

6 References

- [1] JavaMail
<http://java.sun.com/products/javamail/>
- [2] FS Execution Engine
3/FS-MAS0001 Uen
- [3] FS External Component Register
9/FS-MAS0001 Uen
- [4] FS Log Manager
2/FS-MAS0001 Uen

7 Terminology

SMTP

Simple Mail Transfer Protocol