FD – Message Sender

Content

History

Version	Date	Adjustments
PA1	2007-08-28	First version of the document. Original document was found in MAS. (EMAHAGL)

1 Introduction

Message Sender is a backend software component providing an interface for sending MIME messages using SMTP for message submission and transport.

2 Function Structure

2.1 Overview

Message sender has a layered structure. Interface layer, Session layer and Networking layer. All layers accept a RFC822 MIME message implemented as a JavaMail Message.

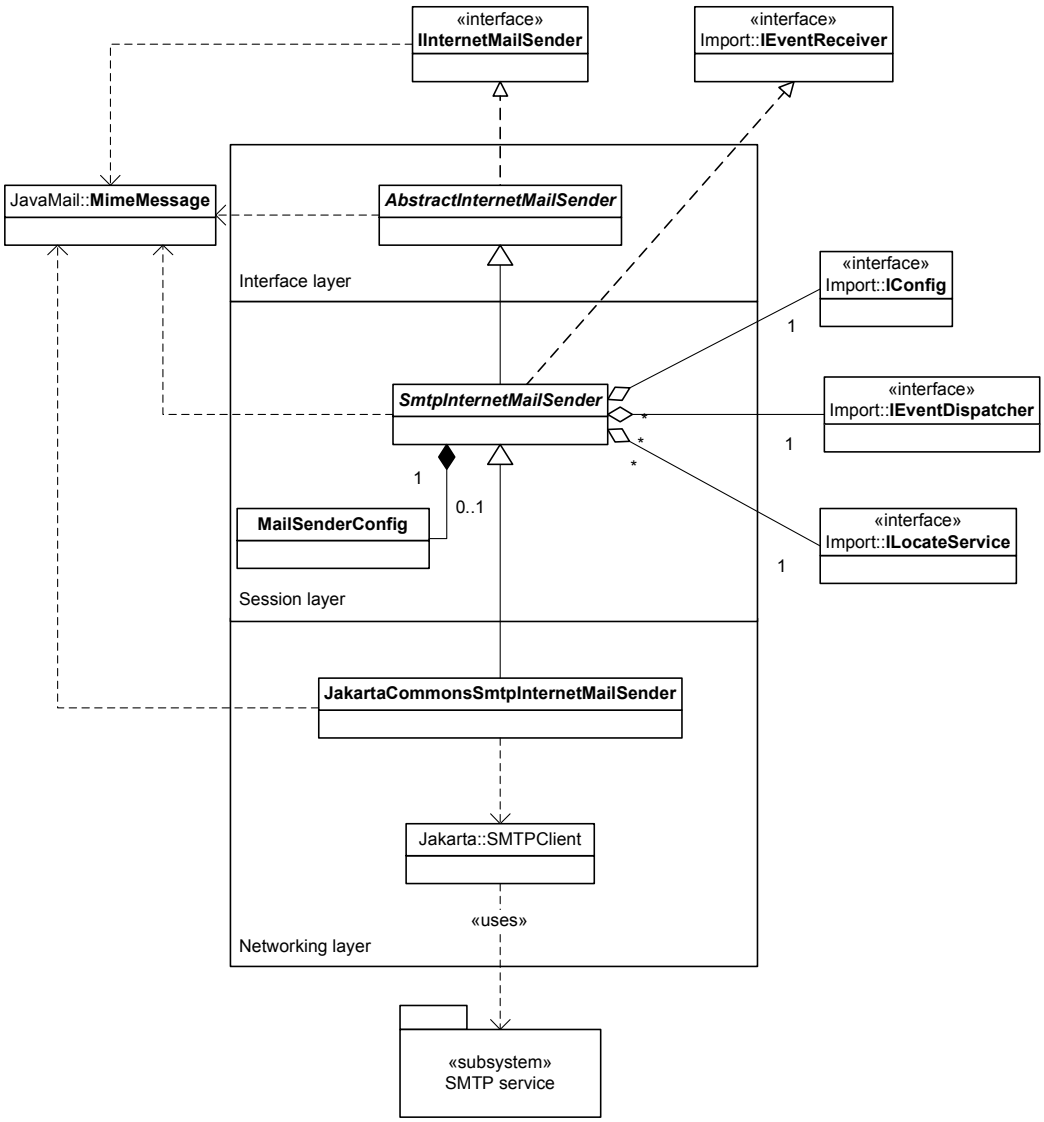


Figure 1 Structure of Message Sender

2.2 Interface layer

The interface layer provides the IInternetMailSender interface. Informational logging of interface method calls and return values is carried out in the interface layer.

2.2.1 Implemented interfaces

- IInternetMailSender

2.3 Session layer

In the session layer accurate system resources (configuration and SMTP service instance) are allocated for every call to the Message Sender.

2.3.1 Implemented interfaces

- IEventReceiver

2.3.2 Static imports/injected dependencies

- IConfig for configuration.
- ILocateService for locating SMTP service and reporting service errors.
- IEventDispatcher for registering IEventReceiver.

2.4 Networking layer

In the Networking layer is the actual SMTP network interaction takes place. The networking layer is implemented with an open source SMTP socket client API.

3 Function Behavior

3.1 Message integrity

The message will in most cases be untouched by the layer and preserved as the caller passed it.

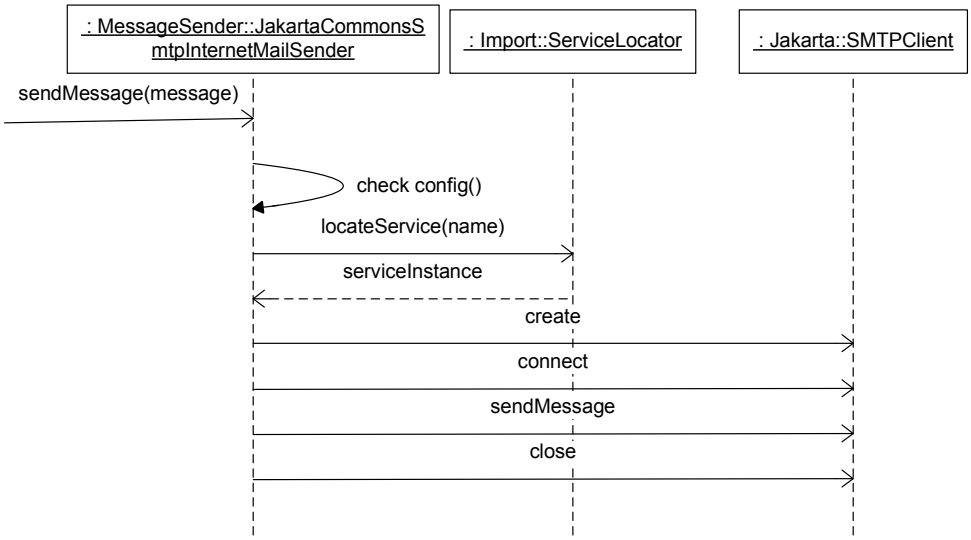
The exception is when a "To" header contains an email address without a domain, e.g. "notification.off". Message sender will add the receiving mail host as domain to such addresses before sending the message. These addresses will also be used as the RCPT TO header in the SMTP envelope.

3.2 Sending Internet mail

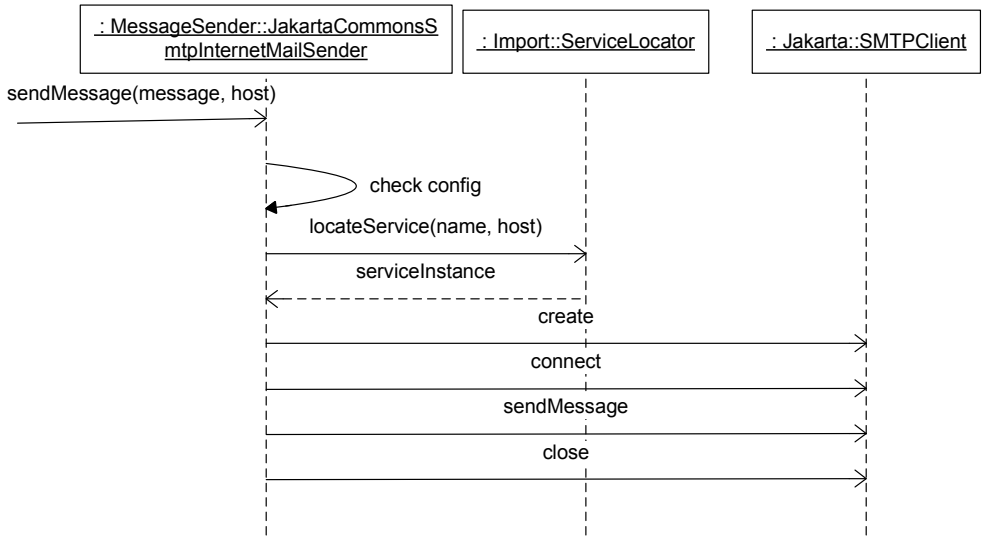
Message Sender receives a MIME message from calling object and deals with the transportation over SMTP to message destination.

The Message Sender basically performs the following steps:

1. Check current configuration.
2. Choose SMTP host.
3. Connect to SMTP service at chosen host.
4. Send message.
5. Close connection.



Sending message with preferred host

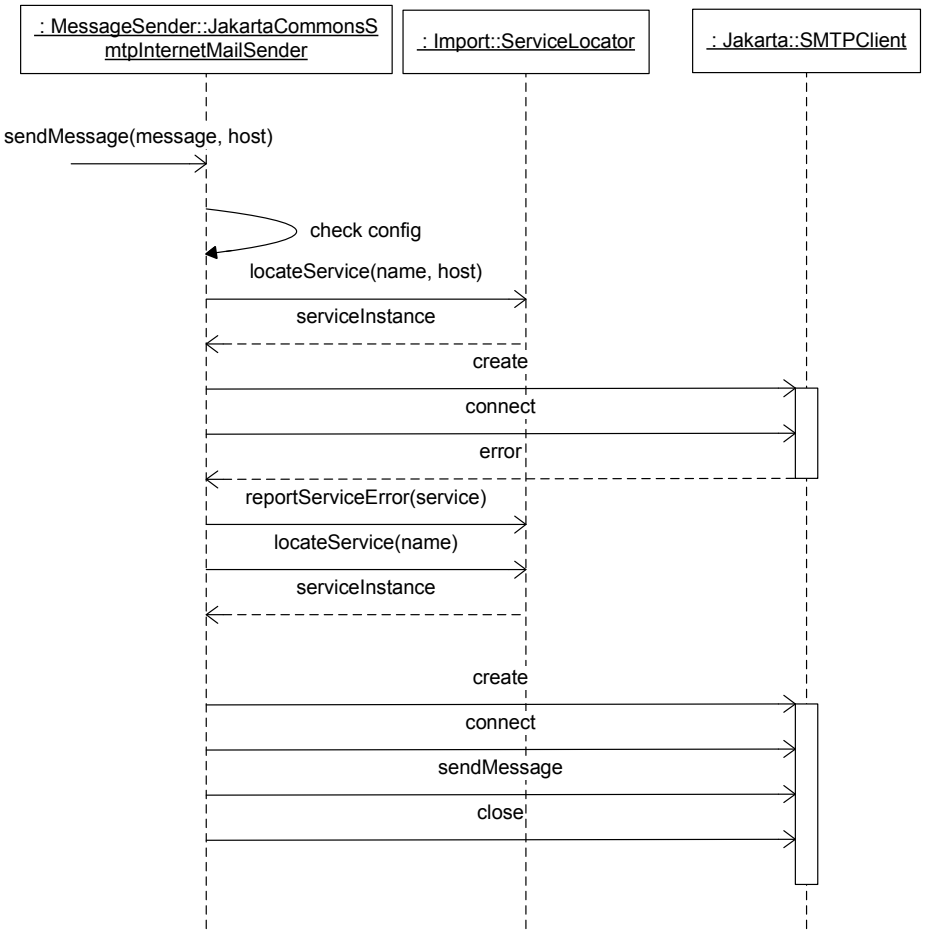


Sending message with given host

Figure 2 Sending a message with preferred and given host

3.2.1 SMTP retries

If error occurs during step 2-4 Message Sender does a configurable number of retries of step 2-4 until step has passed. If none of the retries are successful the error will be propagated back to the calling object.



Sending message with given host, one connection error occurs

Figure 3 Sending a message with retry

3.2.2 Check current configuration

For each call to send a message, Message Sender checks if it has a Configuration object. If not – the latest Message Sender configuration group is fetched from the injected IConfiguration object and a new Configuration object is constructed. The configuration group is then used to initialize the new Configuration object.

3.2.3 Choosing SMTP host

Message Sender needs a SMTP service to be able to send messages. Calling object may explicitly specify the SMTP host to use. Then Message Sender calls the injected ILocateService object to obtain that specific service instance. If no host is specified by calling object Message Sender calls the injected ILocateService object to obtain a preferred host.

3.2.3.1 Retry situation

If error has occurred during an earlier try to send the message and the choosing step is done for a retry – a preferred host is always located no matter if a host was specified by the calling object in the first place.

3.2.4 Connect to SMTP service

Message Sender creates a new SMTP client object and connects to the SMTP service at the chosen host.

3.2.5 Sending message

If the SMTP service connection was successful Message Sender starts the SMTP interaction with the service to send the MIME message.

3.2.5.1 Data terminating dot timeout

The sending step is considered complete if the SMTP data terminating dot is sent. The response from terminating data may timeout. If so - then it is not considered as an error but is Message Sender internally reported as a "likely" success.

3.2.6 Closing connection

Upon a successful termination of data the SMTP service connection is closed.

3.2.6.1 Closing connection timeout

The response from closing the connection may timeout. If so - then it is not considered as an error but is Message Sender internally reported as a "likely" success.

3.2.7 Error handling

There are several possible errors that can occur during the sending process:

- Connection failure due erroneous SMTP service.
- Connection failure due timeout.
- SMTP Command failures due server problems.
- SMTP Command permanent failures.
- SMTP Command failures due timeout.

When an error occurs (in step 2-4) the error is logged and the service instance is reported as erroneous to the injected ILocateService object and retries is performed as described in 3.2.1

3.2.8 Configuration

Message Sender has the below configuration possibilities. Configuration may be changed during runtime. If Message sender receives an event that configuration has changed, it simply removes its Configuration object. Next time step 1 in the sending process is reached the latest configuration will be loaded.

Setting	Description	Default
SMTP Service name	The name to use for locating a SMTP service with the ILocateService	smtpstorage
Connection timeout	Service socket connection timeout value in milliseconds.	3000 ms
SMTP timeout	Socket I/O timeout value in milliseconds.	3000 ms
SMTP retries	Number of attempts to send the message before propagating the errors to calling object	3

3.3 Socket Options

Message Sender has two SMTP socket options. These are implemented as settable properties on the JakartaCommonsSmtpInternetMailSender class.

- SO-Linger time (default value = 5 seconds)
- TCP No delay (default value = false)

4 References

[1] FS MessageSender
5/FS-CRH 109 581-1 Uen

[2] JavaMail
http://java.sun.com/products/javamail/

[3] RFC 822
http://www.ietf.org/rfc/rfc0822.txt

[4] RFC 2821
http://www.ietf.org/rfc/rfc2821.txt

5 Terminology

MIME	Multipurpose Internet Mail Extension
SMTP	Simple Mail Transfer Protocol

6 Appendix B: 3PPs

6.1 Third-Party Products and Freeware

3PPName/ Freeware Name	Version of the product/ freeware	Company	Used for	Delivered with the component	ECCN US/EU	Product No. and R-state
JavaMail API	1.3.3	Sun Microsystem, inc	Hold Mime formatted messages.	Yes	5D002	SWF0004 R1C
JavaBeans Activation Framework	1.0.2	Sun Microsystem, inc	Needed by JavaMail.	Yes	5D002	SWF0007 R1A
Jakarta Commons Net	1.4.1	Apache	Sending Mime messages with SMTP	Yes	EAR99	SWF0066 R1A