


Sending Mails with Attachments

One or more attachments can be included in an e-mail using the MIME Adapter. Each attachment is defined as a single MIME part, which is the basic building block of the MIME format (MIME stands for Multipurpose Internet Mail Extensions). Even the most complicated MIME message boils down to a collection of MIME parts, which can be arranged in a hierarchical structure. The MIME specification refers to messages that contain multiple documents as multipart parts. Once the multipart parts have been created, the SMTP header parameter like MIME version and content type are added before sending the e-mail. The content type is **multipart /mixed**, a multipart type typically used for e-mail attachments.

For more information on the MIME part structure, refer to the documentation of the [MIME Adapter](#).

Example File (Builder project Add-ons/EMail):



<your example path>\Add-ons\EMail\uml\smtpEmailing.xml

On this Page:

- [Example](#)

Related Pages:

- [MIME Adapter](#)

The activity diagram below shows, how attachments are passed to the SMTP server as MIME part input objects of different base types, e.g. String and Blob.

```

graph TD
    Start(( )) --> Task1[1 create an image mime part]
    Task1 --> Task2[2 build the MIME multipart parts]
    Task2 --> Task3[3 create MIME message]
    Task3 --> Task4[4 create input parameters]
    Task4 --> Task5[5 mail the content]
    Task5 --> End(( ))

    Task1 --> GP1[gifPart: MIMEPart]
    Task1 --> TP1[textPart: MIMEPart]
    Task1 --> TAT1[textAttachment: MIMEPart]

    Task2 --> MP1[multipart: Array  
(arrayElement = MIMEPart)]

    Task3 --> C1[content: Blob]
    Task3 --> HP1[headerParameters: Array  
(arrayElement = MIMEHeaderField)]

    Task4 --> S1[sender: String]
    Task4 --> DN1[domainName: String]
    Task4 --> R1[recipients: Array  
(arrayElement = String)]

    Task5 --> SA1[SMTPAdapter]
  
```

1. Action **create an image mime part** transforms the attachments into objects of type **MIMEPart**. If the MIME parts are passed in SOAP requests, they are embedded in the SOAP message body as blobs. When creating the **MIMEPart** objects, you have to make sure to transform the content to a blob if necessary.
2. The MIME parts are stored in the **multipart** array.
3. The action state **create MIME message** has the stereotype **<<MIMEConverter>>** assigned. Its purpose is to assemble the final e-mail content from the multipart array. SMTP header parameters like the MIME version and the content type multipart/mixed will be added to the array **headerParameters**. Along with other input parameters, they will be passed to the SMTP adapter that sends the e-mail.
4. By setting the **contentName**, the text will be attached in a file with the given name – as shown for object **textAttachment**. For object **textPart** no name is given, the text will appear in the mail-body.