

MIME Adapter

MIME (Multi-Purpose Internet Mail Extensions) is an extension of the original Internet e-mail protocol. Using MIME, users can exchange different kinds of data files on the Internet: audio, video, images, application programs, and other kinds, as well as the ASCII text handled in the original protocol, the Simple Mail Transfer Protocol ([SMTP](#)).

Although MIME was designed mainly for SMTP, its use today has grown beyond describing the content of email and now often includes descriptions of [content type](#) in general. In E2E Bridge context, you can use MIME with **SMTP** and **SOAP**.

With the E2E MIME Adapter, you can

- convert a MIME message (header and content) to a multipart structure (**decode**)
- convert a MIME multipart structure to a MIME message (header and content) (**encode**).

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



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

On this Page:

- [The MIME Part Structure](#)
- [Decoding of MIME Messages](#)
- [Encoding of MIME Messages](#)

Related Pages:

- [MIME Adapter Reference](#)
- [SMTP Adapter](#)
- [SOAP Adapter](#)

Related Documentation:

- [Wikipedia: MIME](#)
- [Wikipedia: SMTP](#)

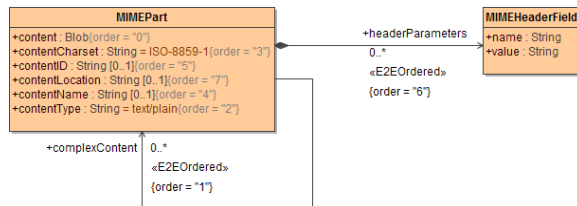
The MIME Part Structure

The content of a MIME part that is provided as input in a SOAP request is base64 binary encoded. The class **MIMEPart** has an attribute **content** of base type **Blob** that stores base64 binary encoded values. If text is used as input for an attachment, the text string needs to be transformed to a blob first, before storing it in a **MIMEPart** object.

The **MIMEPart** class specifies another attribute **contentType**, whose default value is **text/plain**. If it applies to the content type, the **contentCharset** may be defined (default value **ISO-8859-1**). Attribute **contentName** specifies the name of the attachment, e.g. test.txt.

MIME messages can have a hierarchical structure, which can be implemented by assigning an array of **MIMEParts** to the class attribute **complexContent**. Each **MIMEPart** can have zero or many **complexContents**.

Figure: MIME Part Structure

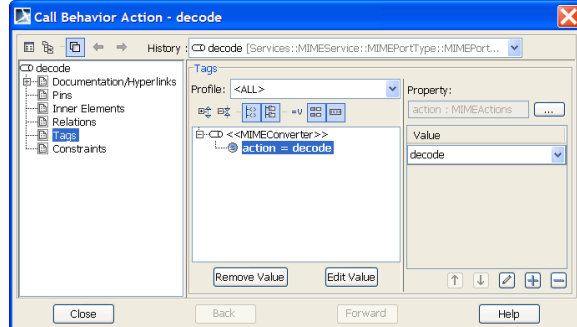


Decoding of MIME Messages

To use the decoding functionality of the E2E MIME Converter:

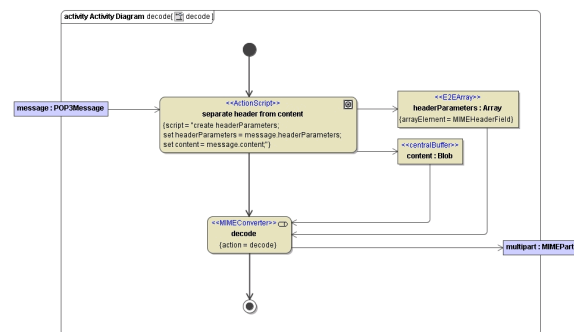
1. Draw an action and stereotype it as **<<MIMEConverter>>**.
2. Open the specification dialog of the action and select **decode** as the value of the tag action.

Figure: Specifying the Tagged Value Decode



In case of decoding, the E2E MIMEConverter takes two input parameters: **headerParameters** and **content**. The first one is an array of header objects of the complex type **MIMEHeaderField**. The second parameter stores the real message content, i.e. the text part of an email as well as the attachments.

Figure: Decoding Example – A Received POP3 Message



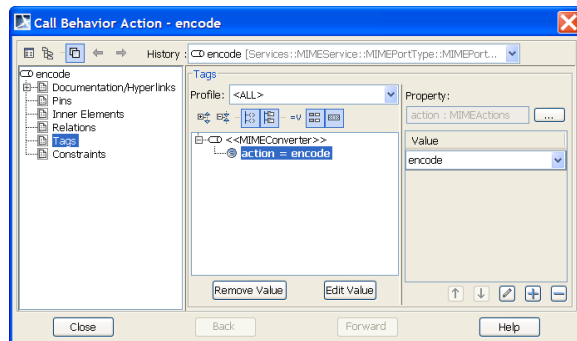
The purpose of the decode action of the MIME converter is to separate the Blob into parts. The parts are encapsulated in output parameter **multipart**, that is of the type **MIMEPart** (as to the class structure refer to the [figure above](#)).

Encoding of MIME Messages

To use the encoding functionality of the E2E MIME Converter:

1. Draw an action and stereotype it as **<<MIMEConverter>>**.
2. Open the specification dialog of the action and select **encode** as the value of the tag action.

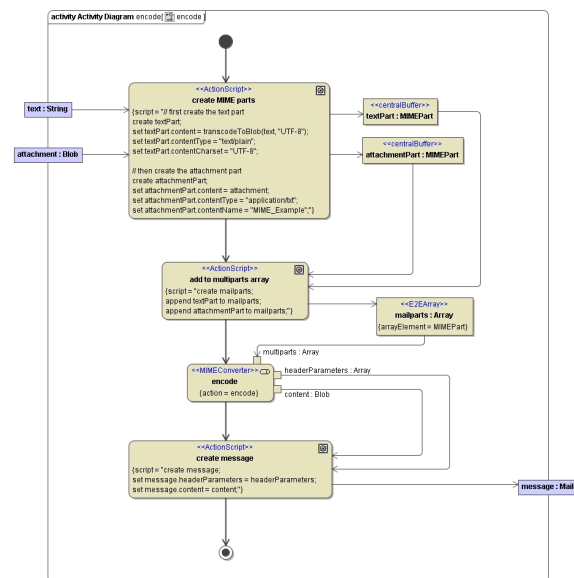
Figure: Specifying the Tagged Value Encode



In the encoding case, the MIME converter takes an array of **MIMEPart** object as a single input parameter. The name of the input parameter must be **multipart/multiparts**. The figure below illustrates how the MIME parts can be created and stored in a single object of type **Array**. Generally, each MIME part represents part of a message (e.g. one **MIMEPart** object stores the plain text of an email message, another the attachment). The content of a **MIMEPart** object is always stored as a **Blob**.

The MIME converter provides two output parameters: **headerParameters** and **content**. The first is an array of **MIMEHeaderField** objects, the latter is a **Blob** containing all the MIME parts.

Figure: Encoding Example – A Content for Mailing



The header parameters and the content can be stored in any object (e.g. **message**) and then passed on to an activity for e.g. sending it over SMTP (for details about how to use the E2E SMTP Adapter refer to [S MTP Adapter](#)).