



This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

# eServices Deployment Guide

Deploying an E-Mail Solution Using MIME Customization

4/21/2025

# Deploying an E-Mail Solution Using MIME Customization

Multipurpose Internet Mail Extensions (MIME) is a standard that allows e-mail messages to include graphics, audio or video files, or text in languages other than English. This section describes how to create a custom transformer that enables you to modify the content of e-mails and still ensure that they are compliant with MIME standards.

MIME customization in a Genesys e-mail handling environment ensures that you have normalized formats for all e-mail messages processed by your system and stored in the UCS database. To accomplish this, your e-mail solution uses a custom transformer to transform any MIME content in incoming or outgoing messages.

You implement this solution by using MIME customization APIs to change the content of incoming and outgoing e-mails. After creating the custom MIME transformer, you configure the E-mail Server Application object to enable the MIME customization option.

Your custom server then does all of the work required to handle incoming and outgoing e-mail messages as needed.

- When an e-mail message is received, E-mail Server uses the custom MIME transformer to transform the message before saving it in the UCS database.
- When an e-mail message is sent, E-mail Server uses the custom MIME transformer to transform the message (if it requires MIME customization) before sending it to the external e-mail server.

The following sections describe how to deploy an E-mail Solution using MIME.

## Deploy MIME Solution

### Deploying a MIME-compliant E-Mail Solution

#### Viewing a sample transformer

E-mail Server includes a sample incoming MIME transformer. This sample is an "MS-TNEF Microsoft specific format to MIME" transformer, and can be found in the `mimeapi` subdirectory.

1. Open the directory where E-mail Server is installed.
2. In the `mimeapi` subdirectory, you can find the following files:

- `esj-mime-api-doc.jar`—Javadoc documentation for the API.
- `esj-mime-api.jar`—The actual API archive.
- `samples`—A subdirectory, containing the following files:
  - `TNEFMimeTransformer.java`—Java source code for this sample.
  - `readme.txt`—A readme file describing the sample.

### Tip

Use the instructions in the `readme.txt` to download the JTNEF library and then compile the source. After you complete these two actions, you can use the sample to transform MS-TNEF incoming mails into regular MIME messages.

## Creating a custom transformer

To transform MIME content of e-mail messages, you must create a custom transformer. Your transformer code needs two Java classes: one to implement the API for transforming MIME content of incoming e-mail, and another to provide an API that transforms MIME content of outgoing e-mail. Both of these classes return the transformation result, and contain the following parameters:

- `input`—The MIME message content.
- `config`—Properties contained in the `[mime-custom-outbound-properties]` section of your E-mail Server Application object.
- `debugLogStream`—Log object to be used for debugging purposes.

Each custom class that you create should satisfy the following conditions:

- Implements one of the two appropriate interfaces.
  - Is thread-safe.
1. Create two Java classes: one for transforming incoming e-mail, another for transforming outgoing e-mail. Use the following interfaces:
    - Incoming E-Mail Customization API
    - Outgoing E-Mail Customization API
  2. Bundle these two classes into a JAR file called `mimecustomization.jar`.
  3. Place the new JAR file in the `esj\lib\external\` folder of your E-mail Server installation.

## Incoming E-Mail Customization API

Your custom transformer for incoming e-mail must implement the following interface:

### EmailInTransformer Interface

```
public interface EmailInTransformer {
    public TransformerResult transform(byte[] input, java.util.Properties config,
        java.io.PrintStream debugLogStream);
    public class TransformerResult {
        public static TransformerResult noTransformationNeeded();
        public static TransformerResult succesfull(byte[] transformedInput);
        public static TransformerResult failure(String failureReason, FailureAction
            failureAction);
        // Implementation details skipped
    }
    public class FailureAction {
        public static final FailureAction RETRY;
        public static final FailureAction BYPASS_TRANSFORMATION;
        public static final FailureAction DEPEND_ON_BAD_FORMAT_OPTION;
        // Implementation details skipped
    }
}
```

## Outgoing E-Mail Customization API

The only difference between this API and the incoming e-mail customization API is that the failureAction parameter and class have been removed. If a transformation fails, then the original message is sent to the external e-mail server. Your custom transformer for outgoing e-mail must implement the following interface:

### EmailOutTransformer Interface

```
public interface EmailOutTransformer {
    public TransformerResult transform(byte[] input, java.util.Properties config,
        java.io.PrintStream debugLogStream);
    public static class TransformerResult {
        public static TransformerResult noTransformationNeeded();
        public static TransformerResult succesfull(byte[] transformedInput);
        public static TransformerResult failure(String failureReason);
        // Implementation details skipped ...
    }
}
```

## Configuring the E-Mail Server Application Object

After you create the custom MIME transformer, configure the E-mail Server Application object to specify the class names and configuration settings used with your custom transformer. E-mail Server will use the values you specify to transform content that is sent and received as e-mail messages. In

Configuration Manager or Genesys Administrator, configure the following sections on the Options panel of your E-mail Server Application object:

- [mime-custom-inbound-properties]—Content in this section is passed to the `EmailInTransformer.transform()` method using the config parameter. You can use this section to define custom options and settings. This section is not part of the default template. Add this section manually to set options for your custom transformer for incoming e-mail.
- [mime-custom-outbound-properties]—Content in this section is passed to the `EmailOutTransformer.transform()` method using the config parameter. You can use this section to define custom options and settings. This section is not part of the default template. Add this section manually to set options for your custom transformer for outgoing e-mail.
- [mime-customization]—Use this section to enable or disable the MIME customization, to specify the fully qualified class names of your custom classes, or to set allow optional debugging or saving features.

## MIME Options

### MIME Customization Options

The following table describes the options for the [mime-customization] Section of the E-mail Server application object.

Option Name	Value	Description
enable-inbound	Default Value: false Valid Values: true, false	If set to true, sends inbound e-mail messages to the specified inbound MIME transformer class.
enable-outbound	Default Value: false Valid Values: true, false	If set to true, sends outgoing e-mail messages to the specified outbound MIME transformer class.
enable-inbound-debug-log	Default Value: false Valid Values: true, false	If set to true, activates the inbound debug logger.
enable-outbound-debug-log	Default Value: false Valid Values: true, false	If set to true, activates the outbound debug logger.
inbound-class-name	Default Value: "" Valid Values: <any string>	Specifies the fully qualified name of the custom inbound transformer.
outbound-class-name	Default Value: "" Valid Values: <any string>	Specifies the fully qualified name of the custom outbound transformer.
inbound-keep-received-mime	Default Value: false	Controls the way E-mail Server saves the content of MIME

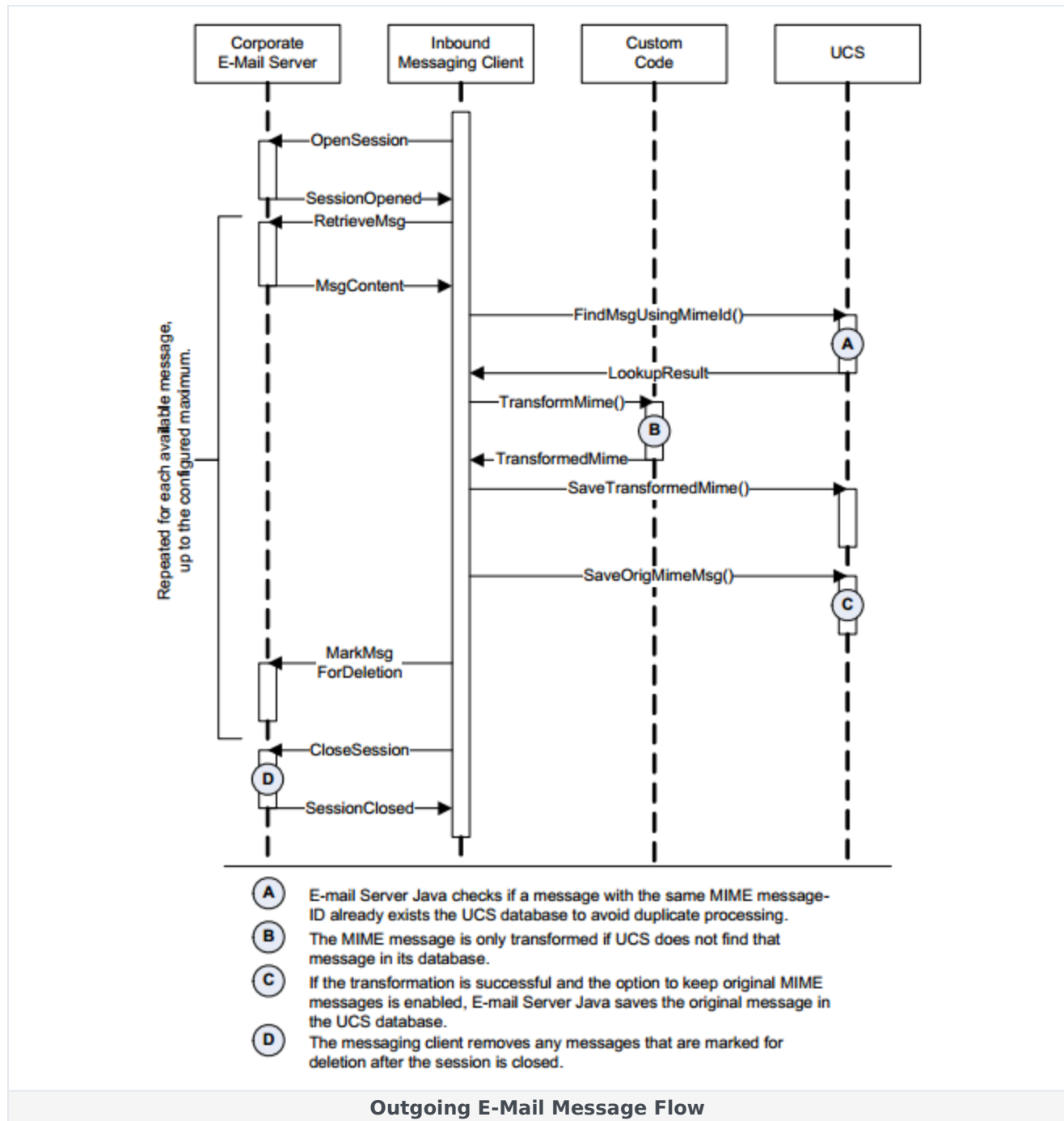
Option Name	Value	Description
	Valid Values: true, false	messages.  If set to true, and if the message was transformed successfully, then saves the unmodified MIME content of incoming e-mails received in the UCS database (along with the transformed content).
outbound-keep-sent-mime	Default Value: false Valid Values: true, false	Controls the way E-mail Server saves the content of MIME messages.  If set to true, and if the message was transformed successfully, then saves the transformed MIME content of outgoing e-mails is also saved in the UCS database (along with the initial content).

## Message Flows

### Message Flow Patterns

The following diagrams show the message patterns for e-mail messages.

#### Incoming E-Mail Message Flow



Outgoing E-Mail Message Flow

