



# Messaging Class

Contains messaging methods used when sending a single or mass email.

## Namespace

[System](#)

## Messaging Methods

The following are methods for `Messaging`. All are instance methods.

- **[extractInboundEmail\(source, includeForwardedAttachments\)](#)**  
Use this method in your email service code to control how to parse and process forwarded or attached emails. Returns an instance of `Messaging.InboundEmail` from a stream of data that is in RFC822 format. The data stream can be a forwarded email in an attachment to an existing `InboundEmail`, or a stream from another source.
- **[reserveMassEmailCapacity\(amountReserved\)](#)**  
Reserves email capacity to send mass email to the specified number of email addresses, after the current transaction commits.
- **[reserveSingleEmailCapacity\(amountReserved\)](#)**  
Reserves email capacity to send single email to the specified number of email addresses, after the current transaction commits.
- **[sendEmail\(emails, allOrNothing\)](#)**  
Sends the list of emails instantiated with either `SingleEmailMessage` or `MassEmailMessage` and returns a list of `SendEmailResult` objects. When org preferences are set to save `EmailMessage` objects and a trigger is defined for `EmailMessage` objects, the trigger is fired for each `SingleEmailMessage` individually. The `sendEmail` method can be called 10 times per Apex transaction and each method invocation can include up to 100 "To", 25 "Cc", and 25 "Bcc" recipients.
- **[sendEmailMessage\(emailMessageIds, allOrNothing\)](#)**  
Sends draft email messages as defined by the specified email message IDs and returns a list of `SendEmailResult` objects.
- **[renderEmailTemplate\(whoId, whatId, bodies\)](#)**  
Replaces merge fields in text bodies of email templates with values from Salesforce records. Returns an array of `RenderEmailTemplateBodyResult` objects, each of which corresponds to an element in the supplied array of text bodies. Each `RenderEmailTemplateBodyResult` provides a success or failure indication, along with either an error code or the rendered text.
- **[renderStoredEmailTemplate\(templateId, whoId, whatId\)](#)**  
Renders a text, custom, HTML, or Visualforce email template that exists in the database into an instance of `Messaging.SingleEmailMessage`. Includes all attachment content in the returned email message.
- **[renderStoredEmailTemplate\(templateId, whoId, whatId, attachmentRetrievalOption\)](#)**  
Renders a text, custom, HTML, or Visualforce email template that exists in the database into an instance of `Messaging.SingleEmailMessage`. Provides options for including attachment metadata only, attachment metadata and content, or excluding attachments.
- **[renderStoredEmailTemplate\(templateId, whoId, whatId, attachmentRetrievalOption, updateEmailTemplateUsage\)](#)**  
Renders a text, custom, HTML, or Visualforce email template that exists in the database



Use this method in your email service code to control how to parse and process forwarded or attached emails. Returns an instance of `Messaging.InboundEmail` from a stream of data that is in RFC822 format. The data stream can be a forwarded email in an attachment to an existing `InboundEmail`, or a stream from another source.

#### Signature

```
public static Messaging.InboundEmail extractInboundEmail(Object source, Boolean
includeForwardedAttachments)
```

#### Parameters

##### *source*

Type: `Object`

An instance of `Messaging.InboundEmail.BinaryAttachment` whose `MimeTypeSubtype` is `message/rfc822` or a `Blob`. If *source* is a `Blob`, then supply a byte array in RFC822 format.

##### *includeForwardedAttachments*

Type: `Boolean`

This parameter controls how attachments to embedded or forwarded emails are handled. Set to `true` to provide all attachments, even attachments in embedded emails in the `binaryAttachments` and `textAttachments` properties of the returned value. Set to `false` to provide only the attachments that are at the top level of the source email.

#### Return Value

Type: `Messaging.InboundEmail`

## reserveMassEmailCapacity(amountReserved)

Reserves email capacity to send mass email to the specified number of email addresses, after the current transaction commits.

#### Signature

```
public Void reserveMassEmailCapacity(Integer amountReserved)
```

#### Parameters

##### *amountReserved*

Type: `Integer`

#### Return Value

Type: `Void`

#### Usage

This method can be called when you know in advance how many addresses emails will be sent to as a result of the transaction. If the transaction would cause the organization to exceed its daily email limit, using this method results in the following error: `System.HandledException: The daily limit for the org would be exceeded by this request.` If the organization doesn't have permission to send API or mass email, using this method results in the following error: `System.NoAccessException: The organization is not permitted to send email.`

## reserveSingleEmailCapacity(amountReserved)

Reserves email capacity to send single email to the specified number of email addresses, after the current transaction commits.



### *amountReserved*

Type: [Integer](#)

### Return Value

Type: Void

### Usage

This method can be called when you know in advance how many addresses emails will be sent to as a result of the transaction. If the transaction would cause the organization to exceed its daily email limit, using this method results in the following error: `System.HandledException: The daily limit for the org would be exceeded by this request.` If the organization doesn't have permission to send API or mass email, using this method results in the following error: `System.NoAccessException: The organization is not permitted to send email.`

## sendEmail(emails, allOrNothing)

Sends the list of emails instantiated with either `SingleEmailMessage` Or `MassEmailMessage` and returns a list of `SendEmailResult` objects. When org preferences are set to save `EmailMessage` objects and a trigger is defined for `EmailMessage` objects, the trigger is fired for each `SingleEmailMessage` individually. The `sendEmail` method can be called 10 times per Apex transaction and each method invocation can include up to 100 "To", 25 "Cc", and 25 "Bcc" recipients.

### Signature

```
public Messaging.SendEmailResult[] sendEmail(Messaging.Email[] emails, Boolean allOrNothing)
```

### Parameters

#### *emails*

Type: [Messaging.Email\[\]](#)

#### *allOrNothing*

Type: [Boolean](#)

The optional `opt_allOrNone` parameter specifies whether `sendEmail` prevents delivery of all other messages when any of the messages fail due to an error (`true`), or whether it allows delivery of the messages that don't have errors (`false`). The default is `true`.

### Return Value

Type: [Messaging.SendEmailResult\[\]](#)

## sendEmailMessage(emailMessageIds, allOrNothing)

Sends draft email messages as defined by the specified email message IDs and returns a list of `SendEmailResult` objects.

### Signature

```
public Messaging.SendEmailResult[] sendEmailMessage(List<ID> emailMessageIds, Boolean allOrNothing)
```

### Parameters

#### *emailMessageIds*

Type: [List<ID>](#)

#### *allOrNothing*

Type: [Boolean](#)



exception.

### Usage

The `sendMessage` method assumes that the optional *allOrNothing* parameter is always `false` and ignores the value you set. Delivery of all messages is attempted even if some messages fail due to an error.

The email address of the user calling the `sendMessage` method is inserted in the From Address field of the email header and the Email Message record.

### Example

This example shows how to send a draft email message. It creates a case and a new email message associated with the case. Next, the example sends a draft email message and checks the results. Before running this example, make sure to replace the email address with a valid address.

```
Case c = new Case();
insert c;

EmailMessage e = new EmailMessage();
e.parentid = c.id;
// Set to draft status.
// This status is required
// for sendMessage().
e.Status = '5';
e.TextBody =
    'Sample email message.';
e.Subject = 'Apex sample';
e.ToAddress = 'customer@email.com';
insert e;

List<Messaging.SendEmailResult>
    results =
        Messaging.sendMessage(new ID[]
            { e.id });

System.assertEquals(1, results.size());
System.assertEquals(true,
    results[0].success);
```

### Versioned Behavior Changes

In API version 54.0 and later, a null *emailMessageIds* parameter results in a `System.IllegalArgumentException` exception. In API version 53.0 and earlier, a null *emailMessageIds* parameter results in an error.

## renderEmailTemplate(whoId, whatId, bodies)

Replaces merge fields in text bodies of email templates with values from Salesforce records. Returns an array of `RenderEmailTemplateBodyResult` objects, each of which corresponds to an element in the supplied array of text bodies. Each `RenderEmailTemplateBodyResult` provides a success or failure indication, along with either an error code or the rendered text.

### Signature

```
public static List<Messaging.RenderEmailTemplateBodyResult> renderEmailTemplate(String whoId,
String whatId, List<String> bodies)
```

### Parameters

#### *whoId*

Type: `String`



Type: [String](#)

Identifies an object in the database like an account or opportunity. The record for that object is read and used in merge field processing.

#### ***bodies***

Type: List<[String](#)>

An array of strings that are examined for merge field references. The corresponding data from the object referenced by the `whoId` or `whatId` replaces the merge field reference.

#### **Return Value**

Type: List<Messaging.RenderEmailTemplateBodyResult>

#### **Usage**

Use this method in situations in which you want to dynamically compose blocks of text that are enriched with data from the database. You can then use the rendered blocks of text to compose and send an email or update a text value in another database record.

Executing the `renderEmailTemplate` method counts toward the SOQL governor limit. The number of SOQL queries that this method consumes is the number of elements in the list of strings passed in the *bodies* parameter.

#### **See Also**

- [Execution Governors and Limits](#)

## **renderStoredEmailTemplate(templateId, whoId, whatId)**

Renders a text, custom, HTML, or Visualforce email template that exists in the database into an instance of `Messaging.SingleEmailMessage`. Includes all attachment content in the returned email message.

#### **Signature**

```
public static Messaging.SingleEmailMessage renderStoredEmailTemplate(String templateId, String whoId, String whatId)
```

#### **Parameters**

##### ***templateId***

Type: [String](#)

An email template that exists in the database, such as text, HTML, custom, and Visualforce templates.

##### ***whoId***

Type: [String](#)

The identifier of an object in the database, typically a contact, lead, or user. The database record for that object is read and used in merge field processing.

##### ***whatId***

Type: [String](#)

Identifies an object in the database, like an account or opportunity. The record for that object is read and used in merge field processing.

#### **Return Value**

Type: [Messaging.SingleEmailMessage](#)



#### See Also

- [Execution Governors and Limits](#)

## renderStoredEmailTemplate(templateId, whold, whatId, attachmentRetrievalOption)

Renders a text, custom, HTML, or Visualforce email template that exists in the database into an instance of `Messaging.SingleEmailMessage`. Provides options for including attachment metadata only, attachment metadata and content, or excluding attachments.

#### Signature

```
public static Messaging.SingleEmailMessage renderStoredEmailTemplate(String templateId, String whoId, String whatId, Messaging.AttachmentRetrievalOption attachmentRetrievalOption)
```

#### Parameters

##### *templateId*

Type: [String](#)

An email template that exists in the database, such as text, HTML, custom, and Visualforce templates.

##### *whold*

Type: [String](#)

The identifier of an object in the database, typically a contact, lead, or user. The database record for that object is read and used in merge field processing.

##### *whatId*

Type: [String](#)

Identifies an object in the database, like an account or opportunity. The record for that object is read and used in merge field processing.

##### *attachmentRetrievalOption*

Type: [Messaging.AttachmentRetrievalOption](#)

Specifies options for including attachments in the `fileAttachments` property of the returned `Messaging.SingleEmailMessage`. Set to one of the [Messaging.AttachmentRetrievalOption](#) values to include attachment metadata only, attachment metadata and content, or to exclude attachments.



#### Note

When the *attachmentRetrievalOption* parameter is *not* set to `NONE`, the `entityAttachments` property of `Messaging.SingleEmailMessage` contains the ID of the Salesforce content objects to attach (`ContentVersion` or `Document`). The `fileAttachments` property contains the IDs of attachments, in addition to all the IDs in the `entityAttachments` property. As a result, the ID values in `entityAttachments` are duplicates of the IDs in the `fileAttachments` property. If you call `renderStoredEmailTemplate()` by passing the `METADATA_WITH_BODY` option, and send the rendered email message, the email will contain duplicate attachments. Before using the returned email message with `sendEmail(emails, allOrNothing)`, you can remove attachments from `fileAttachments` that are duplicated in `entityAttachments`.

#### Return Value



query.

## **renderStoredEmailTemplate(templateId, whoId, whatId, attachmentRetrievalOption, updateEmailTemplateUsage)**

Renders a text, custom, HTML, or Visualforce email template that exists in the database into an instance of `Messaging.SingleEmailMessage`. Provides options for including attachment metadata only, attachment metadata and content, or excluding attachments.

### **Signature**

```
public static Messaging.SingleEmailMessage renderStoredEmailTemplate(String templateId, String whoId, String whatId, Messaging.AttachmentRetrievalOption attachmentRetrievalOption, Boolean updateEmailTemplateUsage)
```

### **Parameters**

#### ***templateId***

Type: [String](#)

An email template that exists in the database, such as text, HTML, custom, and Visualforce templates.

#### ***whoId***

Type: [String](#)

The identifier of an object in the database, typically a contact, lead, or user. The database record for that object is read and used in merge field processing.

#### ***whatId***

Type: [String](#)

Identifies an object in the database, like an account or opportunity. The record for that object is read and used in merge field processing.

#### ***attachmentRetrievalOption***

Type: [Messaging.AttachmentRetrievalOption](#)

Specifies options for including attachments in the `fileAttachments` property of the returned `Messaging.SingleEmailMessage`. Set to one of the [Messaging.AttachmentRetrievalOption](#) values to include attachment metadata only, attachment metadata and content, or to exclude attachments.



#### **Note**

When the `attachmentRetrievalOption` parameter is *not* set to `NONE`, the `entityAttachments` property of `Messaging.SingleEmailMessage` contains the ID of the Salesforce content objects to attach (`ContentVersion` or `Document`). The `fileAttachments` property contains the IDs of attachments, in addition to all the IDs in the `entityAttachments` property. As a result, the ID values in `entityAttachments` are duplicates of the IDs in the `fileAttachments` property. If you call `renderStoredEmailTemplate()` by passing the `METADATA_WITH_BODY` option, and send the rendered email message, the email will contain duplicate attachments. Before using the returned email message with [sendEmail\(emails, allOrNothing\)](#), you can remove attachments from `fileAttachments` that are duplicated in `entityAttachments`.

#### ***updateEmailTemplateUsage***

Type: [Boolean](#)



Type: [Messaging.SingleEmailMessage](#)

Usage

Executing the `renderStoredEmailTemplate` method counts toward the SOQL governor limit as one query.

DID THIS ARTICLE SOLVE YOUR ISSUE?

Let us know so we can improve!

[Share your feedback](#)



DEVELOPER CENTERS

- [Heroku](#)
- [MuleSoft](#)
- [Tableau](#)
- [Commerce Cloud](#)
- [Lightning Design System](#)
- [Einstein](#)
- [Quip](#)

POPULAR RESOURCES

- [Documentation](#)
- [Component Library](#)
- [APIs](#)
- [Trailhead](#)
- [Sample Apps](#)
- [Podcasts](#)
- [AppExchange](#)

COMMUNITY

- [Trailblazer Community](#)
- [Events and Calendar](#)
- [Partner Community](#)
- [Blog](#)
- [Salesforce Admins](#)
- [Salesforce Architects](#)

© Copyright 2025 Salesforce, Inc. [All rights reserved.](#) Various trademarks held by their respective owners. Salesforce, Inc. Salesforce Tower, 415 Mission Street, 3rd Floor, San Francisco, CA 94105, United States

[Privacy Information](#) [Terms of Service](#) [Legal](#) [Use of Cookies](#) [Trust](#) [Cookie Preferences](#)

[✔✕ Your Privacy Choices](#) [Responsible Disclosure](#) [Contact](#)