

# First Class Messaging

## User's Guide

July 25, 2015



Beanstalk

Cloud Solutions LLC

# Table of Contents

<b>Overview .....</b>	<b>3</b>
<b>Installation .....</b>	<b>4</b>
<b>Setup .....</b>	<b>5</b>
Creating Settings.....	5
<i>Specifying the Recipient.....</i>	<i>7</i>
<i>Message Subject / Calendar Subject.....</i>	<i>7</i>
<i>Sender .....</i>	<i>8</i>
<i>CC and BCC Fields.....</i>	<i>9</i>
<i>The Message Body .....</i>	<i>9</i>
<i>Priority .....</i>	<i>10</i>
Optional Sections.....	10
<i>Send Error Alerts.....</i>	<i>10</i>
<i>Email Templates.....</i>	<i>11</i>
<i>Calendar Entries.....</i>	<i>11</i>
<i>Include Attachments.....</i>	<i>12</i>
<b>Example Object - Email Request Object .....</b>	<b>13</b>
<b>Log Files.....</b>	<b>14</b>
Individual Logs – Example 1.....	14
Bulk Logs with Sub-logs – Example 2.....	16
<b>Setting up a Trigger.....</b>	<b>17</b>
Trigger Switches.....	18
<b>Calling from Process Builder or a Flow .....</b>	<b>18</b>
<b>Adding a Send (Resend) Button .....</b>	<b>20</b>
<b>Governor Limits .....</b>	<b>21</b>
<b>Validation.....</b>	<b>22</b>
<b>Troubleshooting .....</b>	<b>23</b>

## Overview

**First Class Messaging** is an application that is intended for use by Salesforce administrators and developers. It makes it a snap to create full-featured automated emails when simple email alerts just don't have enough flexibility to do what is needed.

## Installation

**First Class Messaging** is a tool for developers and administrators.

Installation is a snap. All you have to do is install to find the listing in the Salesforce AppExchange and to click Get It Now.

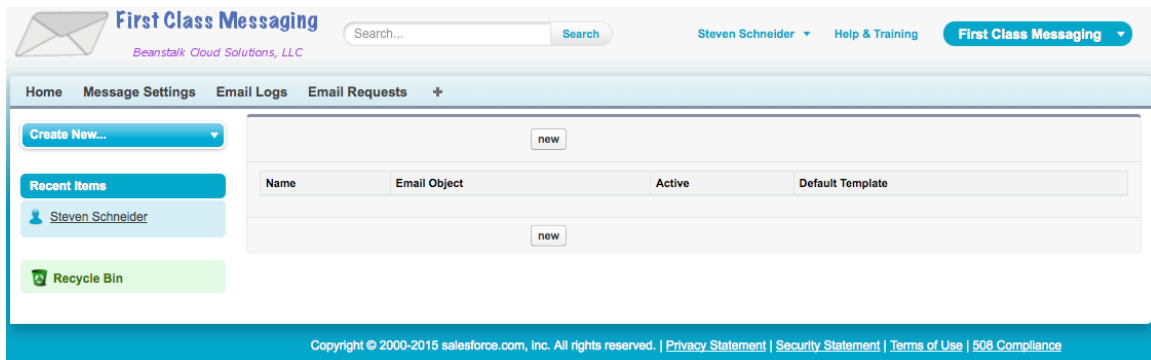


Install into either your production org or a sandbox and install for ***Admin Users Only***.

## Setup

The setup process is intuitive, but it assumes that there is a basic understanding of such things as Salesforce Custom Objects (a.k.a. SObjects) and triggers or Process Builder flows unless you use the out-of-the-box example class to drive email.

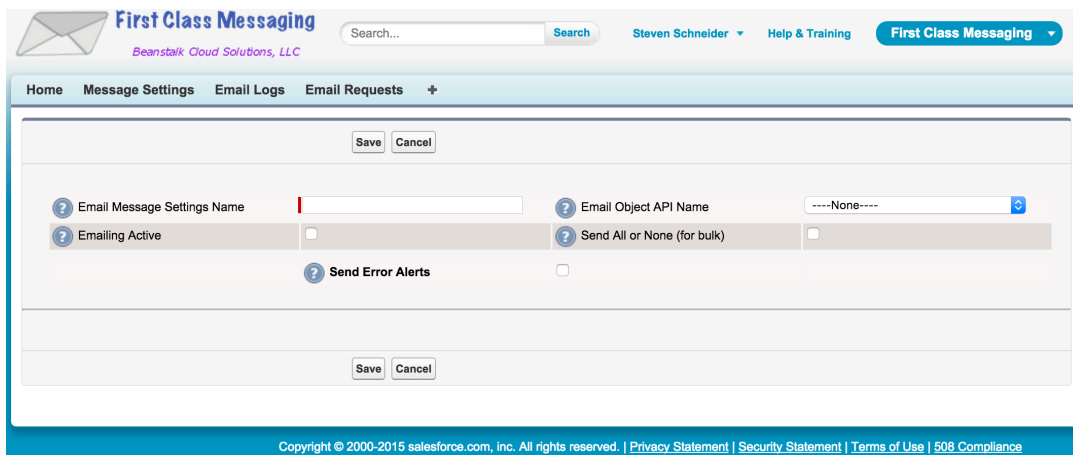
To configure **First Class Messaging** to work with your standard or custom Message objects, you must use the Message Settings wizard to create descriptions of your automated email scenarios.



## Creating Settings

To create message settings, make sure that you are in the **First Class Messaging** application.

1. Click on the Message Settings tab.
2. Click on new.



When the wizard appears, we need to fill in two critical pieces of information:

- **Message Settings Name** The settings name you will see in the settings list.

- **Email Object API Name** The object that drives the emailing. Emails can be triggered by the creation of these objects, and the contents of its fields are used to populate messages.

Two other fields are initially presented that are less critical.

- **Emailing Active** Check this box to turn on emailing for this setting. Provides a way to quickly turn off emailing for all processes that use this setting.
- **Send All or None (for bulk)** When emails are sent in bulk using this setting, should email be sent as all-or-none, or should valid messages be sent while others may fail?

An example Object has been provided that can be used out-of-the-box. It is **Email Request** ( `firstclass_email_request_c` ). This is one object that will be in the drop-down list.

The exception is optional sections that are enabled from checkboxes:

- Send Error Alerts Allows you to specify an email address and/or Chatter group to send error messages to.
- Include Calendar Entry Specify fields necessary to create a calendar attachment (iCalendar - .ics format)
- Include Attachments Several options are available for providing attachments to include with an email message.

Once the email object is selected, most of the other fields appear for defining your email messages. A number of fields have the suffix **API Field Name**. These dropdowns are populated with appropriate fields from the selected Email Object. For example, the **Recipient Name Field API Name** dropdown list is filled with text fields from the Email Object.

- Email Object Id API Field Name The field containing the unique identifier of the object. Usually its value will be “**Id**”.

### Specifying the Recipient

The recipient of a message can either be specified explicitly using the recipient fields below or by specifying a Contact, Lead, or User as a target field.

- Recipient (Target) Name Field API Name The field name from which to pull the name of the person who will be receiving the email message.
- Recipient (Target) Email Field API Name The field name from which to pull the email address of the person who will be receiving the email message.
- Template Target Field API Name Field name for a lookup to a Contact, Lead, or User used as the recipient of the email. This field is required when using a template but can be used without templates to provide the recipient.

### Message Subject / Calendar Subject

Subjects are used for email messages as well as for calendar invitations. To keep things simple and avoid confusion, **First Class Messaging** provides one subject that can be used for both. If Email Templates are used, however, the template furnishes the subject for the email.

- **Subject Field API Name**      The subject field name in the email object used for automated emails. This field may be used as the subject of the email (only when an email template is not used) AND as the subject of the calendar invitation, if one is included. (To keep things simple, calendar invitations use the same subject field).
- **Default Subject**      The default subject for emails if a subject field is not provided. This field may be used as the subject of the email (only when an email template is not used) AND as the subject of the calendar invitation, if one is included.

### Sender

It is preferable to use the Organization-Wide Email Addresses as the options for the sender than to specify other senders, just because that is their intended use – to be the sender email addresses for the organization. If one of these addresses is used, the other email options are hidden from view.

- **Org Wide Default From Email**      The email address from the "Organization-Wide Addresses" from which the message will be sent. The sender name and email address will not be used if this option is used.
- **Sender Name Field API Name**      The sender name field for emails in the email object.
- **Default Sender Name**      Default name of the message Sender used if the field name is not provided.
- **Sender Email Field API Name**      Sender Email Field name in object used for automated emails.
- **Default Sender Email Address**      Default email address of the meeting Sender used if the field is not provided.



### CC and BCC Fields

The CC and BCC fields can be used together to add recipients of the email messages. In other words, all four fields could be populated and could provide recipients for the email messages.

- **CC Email Field API Name** Field name on the email object that houses a list of CC email addresses.
- **Additional CC Emails** Additional email addresses to copy in the CC field of the email messages. All messages sent using this setting would include these explicitly specified email addresses.
- **BCC Email Field API Name** Field name on the email object that houses a list of CC email addresses.
- **Additional BCC Emails** Additional email addresses to copy in the BCC field of the email messages. All messages sent using this setting would include these explicitly specified email addresses.

### The Message Body

When a template is used, the body of the message is that of the template. If a template is not specified or the template is not found, the message body can come from a set of message body fields.

The fields specify the HTML version of the email message and the plain text version. Most email clients allow you to view rich HTML content, although there could be email recipients who can only view plain text. Those people will see the plain text version of the message. Both versions may be supplied to be safe. (This is true for email templates, as well.)

The message body fields are:

- **Message Body HTML Field API Name** Field name in object used to hold an HTML message body that can be used if templates are not selected.
- **Default Message Body HTML** Default message body in HTML format. If this field is provided, plain text message bodies are not shown if recipients' email clients can view

HTML, which most can.

- **Message Body Plain Text Field API Name** Field name in object used to hold a text message body that can be used if templates are not selected.
- **Default Message Body Plain Text** Default plain text message body. If an HTML message body is provided, plain text message bodies are not shown if recipients' email clients can view HTML, which most can.

### Priority

The priority fields are used to specify the importance of the email messages sent using this setting. As with many other fields, the value can be pulled from the email object, or a default value can be used. Values that should be used are:

**Normal** (default), **High**, **Highest**, **Low**, and **Lowest**

- **Priority Field API Name** Message priority field name in object used for automated emails.
- **Default Priority** The default priority of the message

### Optional Sections

The following sections are optionally activated via checkboxes.

#### Send Error Alerts

This checkbox reveals a set of fields for configuring error alerts. If an automated email message fails to be sent, administrators can be notified via email, by a post to a Chatter group, or both. Note that errors relate to the sending of the messages due to problems with the messages, themselves (huge attachments, missing information, etc.) but once the email is sent, it is considered successful.

**First Class Messaging** does not track bounced emails.

- **Alert Email (User preferably)** If an error occurs, send an email message to this address. It's best to select the email address of a User, as the User will be looked up and will receive the message even if the governor limit has been hit for emails sent.

- Alert Chatter Group      If provided, this Chatter group will be sent a post about any errors encountered.

### Email Templates

When the “Use Email Template” checkbox is selected, a set of fields is presented which allow you to control the selection of an email template and the object used to populate “merge” fields, if any are present.

- Template ID Field API Name      Template ID field name in object used for invitations. This controls the template used for the email.
- Default Template      “Template Unique Name” used to find an email template if the template field is not provided.
- Template "Merge Fields" Source      The source of the data used to populate email template "merge fields". Values are
  - Use Email Object
  - Use other lookup object

This field is optional, but it requires that you have a Template Target of type Contact.

- Template Data Source Field API Name      Field name that provides the source of the data used to populate email template "merge fields". This field only appears when the **Template "Merge Fields" Source** is set to “Use other lookup object”.

Note - Only accounts, assets, campaigns, cases, contracts, opportunities, orders, products, solutions and custom objects may be used here. This field is often referred to as the **whatId**.

### Calendar Entries

The “Include Calendar Entry” checkbox, when checked, reveals a set of fields for defining an attached calendar attachment.

- All Day Event Flag Field API Name The field that contains a checkbox for whether or not the meeting or event is all day. If left blank, assumed to never be all day.
- Default All Day Event Flag Check this if the meeting or event lasts all-day
- Start Datetime Field API Name Start Datetime field name in object used for invitations.
- Default Event Start Datetime Default start date and time of an invitation
- Duration Minutes Field API Name Duration in minutes Field name in object used for invitations.
- Default Duration in Minutes Default event duration in minutes if field is not provided.
- Invitation Description Field API Name Event Description Field name used for calendar invitations.
- Default Invitation Description Default calendar invitation event description used if a field is not provided.
- Invitation Event Location Field API Name Event Location Field name used for calendar invitations.
- Default Invitation Event Location The location to be populated in the calendar invitation if a field is not provided.

### Include Attachments

When the **Include Attachments** checkbox is selected, a set of fields appear that allow you to specify where to go to get attachments to include with the email messages. Use this sparingly, as attachments require more system resources to process, potentially reducing the number of messages that can be sent at a time.

- Include Email Object Attachments Include any attachments on the email object with the message. Do not check this unless it is really needed, as it will require additional database activity.

- **Attachment Object Id Field API Name** The Field name containing the Id for the object whose attachments will be included. The Id can be of any type. Do not check this unless it is really needed, as it will require additional database activity.

## Example Object - Email Request Object

**First Class Messaging** can be used with any Standard or Custom object, but an example object has been provided and can be used out of the box, if desired.

The Object definition is below. You can use this as a reference for building similar objects. Not all fields are required. Settings, described later, can supplement the request with default values, and some features may not be needed, depending on your requirements.

### Example SOBJect: Email Request (Email\_Request\_\_c)

Field Label	API Name	Data Type
Attachment Object Id	FirstClass__Attachment_Object_Id__c	Text(25)
BCC Email Addresses	FirstClass__BCC_Email_Addresses__c	Text(255)
CC Email Addresses	FirstClass__CC_Email_Addresses__c	Text(255)
Invitation All Day Event Flag	FirstClass__Invitation_All_Day_Event_Flag__c	Checkbox
Invitation Description	FirstClass__Invitation_Description__c	Text Area(255)
Invitation Duration in Minutes	FirstClass__Invitation_Duration_in_Minutes__c	Number(18, 0)
Invitation Event Date and Time	FirstClass__Invitation_Event_Date_and_Time__c	Date/Time
Invitation Event Location	FirstClass__Invitation_Event_Location__c	Text(255)
Message Body HTML	FirstClass__Message_Body_HTML__c	Long Text Area(32768)
Message Body Plain Text	FirstClass__Message_Body_Plain_Text__c	Long Text Area(32768)
Message Subject	FirstClass__Message_Subject__c	Text(255)
Priority	FirstClass__Priority__c	Picklist (Highest, High, Normal, Low, Lowest )
Sender Email	FirstClass__Sender_Email__c	Email
Sender Name	FirstClass__Sender_Name__c	Text(255)
Target Contact	FirstClass__Target_Contact__c	Lookup(Contact)
Target Lead	FirstClass__Target_Lead__c	Lookup(Lead)
Target User	FirstClass__Target_User__c	Lookup(User)
Target Email Address	FirstClass__Target_Email_Address__c	Email
Target Name	FirstClass__Target_Name__c	Text(255)
Template Unique Name	FirstClass__Template_Unique_Name__c	Text(255)
Template Merge Source Id (What Id) String	FirstClass__Template_Merge_Source_Id_String__c	Text(255)

One message setting should be active for a given SObject. The reason is that a typical trigger will be set to automatically call the email routine, and it looks up the first active setting for the SObject.

## More settings are allowed, but make sure that you are aware that a trigger could pick the wrong setting, depending on how it is implemented. See Log Files

Part of the power of **First Class Messaging** is the logging capabilities. Each message is logged, as are sets of messages, if there are multiple messages sent in bulk. In fact, bulk messages have logs and sub-logs that are connected.

### Individual Logs – Example 1

Below is an example of a successful message.

The screenshot displays the 'Email Logs' interface for a specific message, MSG-10005. The interface includes a header with 'Email Logs' and 'Email Requests' tabs, and a sub-header for the selected log. Below this, there are several sections: 'Email Log Detail' with fields for Log Type Summary, Email Log Number, Email Setting Name, Target Email Address, Template Name, Template Population Object Id, and Source SObject Id String; 'Was Sent' status; 'Created By' and 'Last Modified By' information; and a 'Log Details' section containing a series of debug logs from the FCM\_MessageSender class, detailing the process of creating and sending the email, including attachment handling and recipient information.

If the message fails to be sent, you may see a message such as the one that follows.

Email Logs (0) | Open Activities (0) | Activity History (0)

### Email Log Detail

[Edit](#) [Delete](#) [Clone](#) [Resend Message\(s\)](#) [Resend Failed Messages Only](#)

Log Type Summary This is a log for a single email that was sent.

Email Log Number	MSG-10002	Owner	Steven Schneider <a href="#">[Change]</a>
Email Setting Name	templateTests	Sent Status	Failed
Target Email Address		Status Message	Error executing query: No such column 'Template_ID__c' on entity 'FirstClass__Email_Request__c'. If you are attempting to use a custom field, be sure to append the '__c' after the custom field name. Please reference your WSDL or the describe call for the
Template Name		Target Object Type	
Template Population Object Id		Target Email Object Id	
Template Population Object Type		Parent Email Log	
Source SObject Id String	a01i000000UQUzAAH	Has Children	<input type="checkbox"/>
Was Sent	<input type="checkbox"/>		
Created By	Steven Schneider, 5/13/2015 10:40 PM	Last Modified By	Steven Schneider, 5/13/2015 10:40 PM

Log Details [\[DEBUG\]](#) Email Sender : FCM\_MessageSender : query: Select Invitation\_\_All\_Day\_Event\_Flag\_\_c, Invitation\_Duration\_in\_Minutes\_\_c, Invitation\_Description\_\_c, Sender\_Email\_\_c, Sender\_Name\_\_c, Invitation\_Event\_Date\_and\_Time\_\_c, Message\_Subject\_\_c, Template\_ID\_\_c, Target\_Email\_Address\_\_c, Name, Message\_Body\_Plain\_Text\_\_c, Message\_Body\_HTML\_\_c, Target\_User\_\_c, Priority\_\_c, Attachment\_Object\_Id\_\_c, Id from firstclass\_\_email\_request\_\_c where Id IN :ids  
[\[DEBUG\]](#) Email Sender : FCM\_MessageSender : ids: {a01i000000UQUzAAH}  
[\[DEBUG\]](#) Email Sender : FCM\_MessageSender : Error executing query: No such column 'Template\_ID\_\_c' on entity 'FirstClass\_\_Email\_Request\_\_c'. If you are attempting to use a custom field, be sure to append the '\_\_c' after the custom field name. Please reference your WSDL or the describe call for the appropriate names.  
Query: Select Invitation\_\_All\_Day\_Event\_Flag\_\_c, Invitation\_Duration\_in\_Minutes\_\_c, Invitation\_Description\_\_c, Sender\_Email\_\_c, Sender\_Name\_\_c, Invitation\_Event\_Date\_and\_Time\_\_c, Message\_Subject\_\_c, Template\_ID\_\_c, Target\_Email\_Address\_\_c, Name, Message\_Body\_Plain\_Text\_\_c, Message\_Body\_HTML\_\_c, Target\_User\_\_c, Priority\_\_c, Attachment\_Object\_Id\_\_c, Id from firstclass\_\_email\_request\_\_c where Id IN :ids

[Edit](#) [Delete](#) [Clone](#) [Resend Message\(s\)](#) [Resend Failed Messages Only](#)

Note that the log is really just a snippet of a typical Salesforce debug log. It shows the relevant information for the individual message, however.

This is where Salesforce Administration or Development skills can come in handy.

In this case, it reveals the database query used to fetch data from an *Email Object*. A field that had been specified does not seem to exist. (This can happen if the object was edited to rename or remove any fields after setting up an email setting.)

### Bulk Logs with Sub-logs – Example 2

If email is sent from several email objects at once, **First Class Messaging**'s email routine is called with several object Ids at once. When this happens, a series of log messages are created, one parent log with child logs.

The image that follows shows a parent log. It has a summary at the top stating that it is a parent log, and it has a checkbox on the right side showing that it has children. There is also a related list of the children. If you click on a child record, it takes you to another screen showing its details.

Notice that the parent log gives a status summary saying that all children failed. This is due to an error of having hit the governor limit for total emails in a day.

[Email Logs \(2\)](#) | [Open Activities \(0\)](#) | [Activity History \(0\)](#)

**Email Log Detail**

[Edit](#) [Delete](#) [Clone](#) [Resend Message\(s\)](#) [Resend Failed Messages Only](#)

Log Type Summary This is a parent log. You can see details of individual emails by using the Email Log related list below.

Email Log Number	MSG-08694	Owner	<a href="#">Steven Schneider</a> <a href="#">[Change]</a>
Email Setting Name	templateTests	Sent Status	All messages failed
Target Email Address		Status Message	All 2 messages failed.
Template Name		Target Object Type	
Template Population Object Id		Target Email Object Id	
Template Population Object Type		Parent Email Log	
Source SObject Id String		Has Children	<input checked="" type="checkbox"/>
Was Sent	<input type="checkbox"/>		
Created By	<a href="#">Steven Schneider</a> , 4/14/2015 10:52 PM	Last Modified By	<a href="#">Steven Schneider</a> , 4/14/2015 10:52 PM

Log Details [DEBUG] Email Sender : FCM\_MessageSender : query: Select Sender\_Email\_\_c, Sender\_Name\_\_c, Message\_Subject\_\_c, Template\_ID\_\_c, Target\_Err  
 Target\_Contact\_\_c, Priority\_\_c, Attachment\_Object\_Id\_\_c, Id from firstclass\_\_email\_request\_\_c where Id IN :ids  
 [DEBUG] Email Sender : FCM\_MessageSender : ids: {a01000000UQJazAAH, a01000000WINKbAAN}  
 [DEBUG] Email Sender : FCM\_MessageSender : Adding email attachments for this email descriptor (id: a01000000WINKbAAN ).  
 [DEBUG] Email Sender : FCM\_MessageSender : Adding email attachments for this email descriptor (id: 003i000000NhD4VAAV ).  
 [DEBUG] Email Sender : FCM\_MessageSender : Adding email attachments for this email descriptor (id: a01000000UQJazAAH ).  
 [DEBUG] Email Sender : FCM\_MessageSender : Adding email attachments for this email descriptor (id: 003i000000NhD4VAAV ).  
 [DEBUG] Email Sender : FCM\_MessageSender : Attachment query: select Id, BodyLength, ParentId, IsPrivate, IsDeleted from Attachment where ParentId  
 (a01000000WINKbAAN, 003i000000NhD4VAAV, a01000000UQJazAAH, 003i000000NhD4VAAV)  
 [DEBUG] Email Sender : FCM\_MessageSender : Remaining heap (- 800K cushion): 5166237. Total attachments: 2657768  
 [DEBUG] Email Sender : FCM\_MessageSender : Exception sending email: The daily limit for the org would be exceeded by this request

[Edit](#) [Delete](#) [Clone](#) [Resend Message\(s\)](#) [Resend Failed Messages Only](#)

Action	Email Log Number
<a href="#">Edit</a>   <a href="#">Del</a>	<a href="#">MSG-08695</a>
<a href="#">Edit</a>   <a href="#">Del</a>	<a href="#">MSG-08696</a>

The log file below is a child of the parent we just looked at. Its status message states that there was an error with the parent (the governor limit) that caused this message to fail.

[Email Logs \(2\)](#) | [Open Activities \(0\)](#) | [Activity History \(0\)](#)

**Email Log Detail**

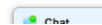
[Edit](#) [Delete](#) [Clone](#) [Resend Message\(s\)](#) [Resend Failed Messages Only](#)

Log Type Summary This is a child log for a single email message. Other emails were sent. To see them, go to the parent email log to get to any child.

Email Log Number	MSG-08695	Owner	<a href="#">Steven Schneider</a> <a href="#">[Change]</a>
Email Setting Name	templateTests	Sent Status	Failed
Target Email Address		Status Message	Parent failure: Exception sending email: The daily limit for the org would be exceeded by this request
Template Name		Target Object Type	
Template Population Object Id		Target Email Object Id	
Template Population Object Type		Parent Email Log	<a href="#">MSG-08694</a>
Source SObject Id String	a01000000WINKbAAN	Has Children	<input type="checkbox"/>
Was Sent	<input type="checkbox"/>		
Created By	<a href="#">Steven Schneider</a> , 4/14/2015 10:52 PM	Last Modified By	<a href="#">Steven Schneider</a> , 4/14/2015 10:52 PM

Log Details [DEBUG] Email Sender : FCM\_MessageSender : Read String Field ( Id ) returns: a01000000WINKbAAN  
 [DEBUG] Email Sender : FCM\_MessageSender : Read String Field ( null ) returns: null  
 [DEBUG] Email Sender : FCM\_MessageSender : Read String Field ( Sender\_Email\_\_c ) returns: [synthought@gmail.com](#)  
 [DEBUG] Email Sender : FCM\_MessageSender : Read String Field ( Sender\_Name\_\_c ) returns: Steve Sender  
 [DEBUG] Email Sender : FCM\_MessageSender : Read String Field ( null ) returns: null  
 [DEBUG] Email Sender : FCM\_MessageSender : Read String Field ( Priority\_\_c ) returns: Highest  
 [DEBUG] Email Sender : FCM\_MessageSender : Read String Field ( Message\_Subject\_\_c ) returns: big attachment  
 [DEBUG] Email Sender : FCM\_MessageSender : Read String Field ( Template\_ID\_\_c ) returns: First\_Class\_Messagin...  
 [DEBUG] Email Sender : FCM\_MessageSender : Being sent to email address: null  
 [DEBUG] Email Sender : FCM\_MessageSender : Read String Field ( Name ) returns: ERN-9395  
 [DEBUG] Email Sender : FCM\_MessageSender : To Email Address: null toName: ERN-9395  
 [DEBUG] Email Sender : FCM\_MessageSender : What Id (merge data object): a01000000WINKbAAN  
 [DEBUG] Email Sender : FCM\_MessageSender : Error: There was an attachment that was larger than the max size.

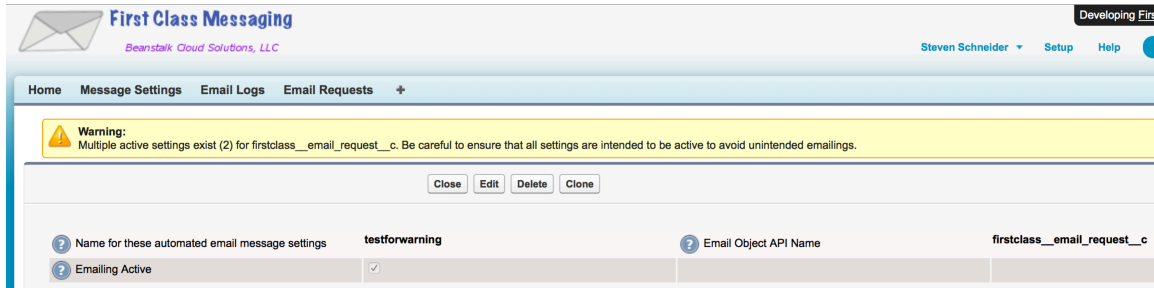
[Edit](#) [Delete](#) [Clone](#) [Resend Message\(s\)](#) [Resend Failed Messages Only](#)





The log messages also have buttons to resend a message or just failed messages. When clicked at the parent level, a bulk set of messages can be resent or just failed messages. This can be particularly useful in cases like this where there might not be anything wrong with the messages, themselves, but we have simply hit a governor limit.

Setting up a Trigger.



## Log Files

Part of the power of **First Class Messaging** is the logging capabilities. Each message is logged, as are sets of messages, if there are multiple messages sent in bulk. In fact, bulk messages have logs and sub-logs that are connected.

### Individual Logs – Example 1

Below is an example of a successful message.

Email Logs | Email Requests +

MSG-10005

Back to List: Email Logs

Email Logs | Open Activities | Activity History

**Email Log Detail**

Log Type Summary This is a log for a single email that was sent.

Email Log Number: MSG-10005  
 Email Setting Name: templateTests  
 Target Email Address: [redacted]  
 Template Name: 00X0000000M4mEAE  
 Template Population Object Id: [redacted]  
 Template Population Object Type: [redacted]  
 Source SObject Id String: a01000000QUzAAH

Owner: Steven Schneider (Change)  
 Sent Status: Sent  
 Status Message: Message sent successfully  
 Target Object Type: [redacted]  
 Target Email Object Id: 00Q000000vcvGbEAI  
 Parent Email Log: [redacted]  
 Has Children: [redacted]

Was Sent:   
 Created By: Steven Schneider, 5/13/2015 11:18 PM  
 Last Modified By: Steven Schneider, 5/13/2015 11:18 PM

Log Details

```
[DEBUG] Email Sender: FCM_MessageSender: Setting name: templateTests
[DEBUG] Email Sender: FCM_MessageSender: query: Select Invitation_All_Day_Event_Flag__c, Invitation_Duration_in_Minutes__c, Invitation_Description__c, Sender_Email__c, Sender_Name__c, Invitation_Event_Date_and_Time__c, Message_Subject__c, Target_Email_Address__c, Name, Message_Body_Plain_Text__c, Message_Body_HTML__c, Target_Lead__c, Priority__c, Template_Merge_Source_Id_String__c, Attachment_Object_Id__c, Id from firstclass_email_request__c where Id IN :ids
[DEBUG] Email Sender: FCM_MessageSender: ids: (a01000000QUzAAH)
[DEBUG] Email Sender: FCM_MessageSender: Read String Field ( Id ) returns: a01000000QUzAAH
[DEBUG] Email Sender: FCM_MessageSender: Read String Field ( Invitation_Description__c ) returns: this is a test
[DEBUG] Email Sender: FCM_MessageSender: Read String Field ( Sender_Email__c ) returns: gytbought@gmail.com
[DEBUG] Email Sender: FCM_MessageSender: Read String Field ( Sender_Name__c ) returns: Steve Sender
[DEBUG] Email Sender: FCM_MessageSender: Read String Field ( Message_Body_Plain_Text__c ) returns: This is a test of a ...
[DEBUG] Email Sender: FCM_MessageSender: Read String Field ( Message_Body_HTML__c ) returns: <strong>this is </strong>
[DEBUG] Email Sender: FCM_MessageSender: Read String Field ( Message_Subject__c ) returns: test email subject
[DEBUG] Email Sender: FCM_MessageSender: Being sent to email address: null
[DEBUG] Email Sender: FCM_MessageSender: Remaining heap (- 600K cushion): 5167942. Total attachments: 2657768
[DEBUG] Email Sender: FCM_MessageSender: To Address: null toName: ERN-4948
[DEBUG] Email Sender: FCM_MessageSender: Adding email attachments for this email descriptor (Id: a01000000QUzAAH ).
[DEBUG] Email Sender: FCM_MessageSender: Attachment query: select Id, BodyLength, ParentId, IsPrivate, IsDeleted from Attachment where ParentId in (a01000000QUzAAH, 003000000hD4VAAV)
[DEBUG] Email Sender: FCM_MessageSender: No start time so no attachment was created.
[DEBUG] Email Sender: FCM_MessageSender: Creating Email...set attachments: (Messaging.EmailFileAttachment{getBody:common.udd.object.EncryptableFxBlobFieldsDeferredEncryptableFxBlobValueImpi@4027752c; getContentType:image/jpeg;getFileName=Beanstalk 2.jpg;getInline=false}; Messaging.EmailFileAttachment{getBody:common.udd.object.EncryptableFxBlobFieldsDeferredEncryptableFxBlobValueImpi@876791b3; getContentType:image/jpeg;getFileName=Beanstalk 1.jpg;getInline=false}; Messaging.EmailFileAttachment{getBody:common.fermanager.ByteBlobValue@2c1d3f3;getContentType:image/jpeg;getFileName=team-1.jpg;getInline=false;})
[DEBUG] Email Sender: FCM_MessageSender: Creating Email with recipient email address: null, Target Id: 00Q000000vcvGbEAI, To Addresses: ()
[DEBUG] Email Sender: FCM_MessageSender: Creating Email...templateId: 00X0000000M4mEAE
[DEBUG] Email Sender: FCM_MessageSender: Message sent successfully
```

If the message fails to be sent, you may see a message such as the one that follows.

Email Log

MSG-10002

Back to List: Email Logs

Email Logs | Open Activities | Activity History

**Email Log Detail**

Log Type Summary This is a log for a single email that was sent.

Email Log Number: MSG-10002  
 Email Setting Name: templateTests  
 Target Email Address: [redacted]

Owner: Steven Schneider (Change)  
 Sent Status: Failed  
 Status Message: Error executing query: No such column 'Template\_ID\_\_c' on entity 'FirstClass\_Email\_Request\_\_c'. If you are attempting to use a custom field, be sure to append the '\_\_c' after the custom field name. Please reference your WSDL or the describe call for the

Target Object Type: [redacted]  
 Target Email Object Id: [redacted]  
 Parent Email Log: [redacted]  
 Has Children: [redacted]

Was Sent:   
 Created By: Steven Schneider, 5/13/2015 10:40 PM  
 Last Modified By: Steven Schneider, 5/13/2015 10:40 PM

Log Details

```
[DEBUG] Email Sender: FCM_MessageSender: query: Select Invitation_All_Day_Event_Flag__c, Invitation_Duration_in_Minutes__c, Invitation_Description__c, Sender_Email__c, Sender_Name__c, Invitation_Event_Date_and_Time__c, Message_Subject__c, Template_ID__c, Target_Email_Address__c, Name, Message_Body_Plain_Text__c, Message_Body_HTML__c, Target_User__c, Priority__c, Attachment_Object_Id__c, Id from firstclass_email_request__c where Id IN :ids
[DEBUG] Email Sender: FCM_MessageSender: ids: (a01000000QUzAAH)
[DEBUG] Email Sender: FCM_MessageSender: Error executing query: No such column 'Template_ID__c' on entity 'FirstClass_Email_Request__c'. If you are attempting to use a custom field, be sure to append the '__c' after the custom field name. Please reference your WSDL or the describe call for the appropriate names.
Query: Select Invitation_All_Day_Event_Flag__c, Invitation_Duration_in_Minutes__c, Invitation_Description__c, Sender_Email__c, Sender_Name__c, Invitation_Event_Date_and_Time__c, Message_Subject__c, Template_ID__c, Target_Email_Address__c, Name, Message_Body_Plain_Text__c, Message_Body_HTML__c, Target_User__c, Priority__c, Attachment_Object_Id__c, Id from firstclass_email_request__c where Id IN :ids
```

Note that the log is really just a snippet of a typical Salesforce debug log. It shows the relevant information for the individual message, however.

This is where Salesforce Administration or Development skills can come in handy.

In this case, it reveals the database query used to fetch data from an *Email Object*. A field that had been specified does not seem to exist. (This can happen if the object was edited to rename or remove any fields after setting up an email setting.)

## Bulk Logs with Sub-logs – Example 2

If email is sent from several email objects at once, **First Class Messaging's** email routine is called with several object Ids at once. When this happens, a series of log messages are created, one parent log with child logs.

The image that follows shows a parent log. It has a summary at the top stating that it is a parent log, and it has a checkbox on the right side showing that it has children. There is also a related list of the children. If you click on a child record, it takes you to another screen showing its details.

Notice that the parent log gives a status summary saying that all children failed. This is due to an error of having hit the governor limit for total emails in a day.

**Email Log**  
MSG-08694

Customize Page | Edit Layout | Printable

← Back to List: Email Logs

Email Logs [2] | Open Activities [0] | Activity History [0]

**Email Log Detail** [Edit] [Delete] [Clone] [Resend Message(s)] [Resend Failed Messages Only]

Log Type Summary This is a parent log. You can see details of individual emails by using the Email Log related list below.

Email Log Number	MSG-08694	Owner	Steven Schneider [Change]
Email Setting Name	templateTests	Sent Status	All messages failed
Target Email Address		Status Message	All 2 messages failed.
Template Name		Target Object Type	
Template Population Object Id		Target Email Object Id	
Template Population Object Type		Parent Email Log	
Source SObject Id String		Has Children	<input checked="" type="checkbox"/>
Was Sent	<input type="checkbox"/>		
Created By	Steven Schneider, 4/14/2015 10:52 PM	Last Modified By	Steven Schneider, 4/14/2015 10:52 PM

Log Details [DEBUG] Email Sender : FCM\_MessageSender : query: Select Sender\_Email\_\_c, Sender\_Name\_\_c, Message\_Subject\_\_c, Template\_ID\_\_c, Target\_Err Target\_Contact\_\_c, Priority\_\_c, Attachment\_Object\_Id\_\_c, Id from firstclass\_email\_request\_\_c where Id IN :ids  
 [DEBUG] Email Sender : FCM\_MessageSender : ids: (a01000000UQUzAAH, a01000000WINKbAAN)  
 [DEBUG] Email Sender : FCM\_MessageSender : Adding email attachments for this email descriptor (id: a01000000WINKbAAN ).  
 [DEBUG] Email Sender : FCM\_MessageSender : Adding email attachments for this email descriptor (id: 003000000NhD4VAAV ).  
 [DEBUG] Email Sender : FCM\_MessageSender : Adding email attachments for this email descriptor (id: a01000000UQUzAAH ).  
 [DEBUG] Email Sender : FCM\_MessageSender : Adding email attachments for this email descriptor (id: 003000000NhD4VAAV ).  
 [DEBUG] Email Sender : FCM\_MessageSender : Attachment query: select Id, BodyLength, ParentId, IsPrivate, IsDeleted from Attachment where ParentId (a01000000WINKbAAN, 003000000NhD4VAAV, a01000000UQUzAAH, 003000000NhD4VAAV)  
 [DEBUG] Email Sender : FCM\_MessageSender : Remaining heap (- 800K cushion): 5166237. Total attachments: 2657768  
 [DEBUG] Email Sender : FCM\_MessageSender : Exception sending email: The daily limit for the org would be exceeded by this request

[Edit] [Delete] [Clone] [Resend Message(s)] [Resend Failed Messages Only]

**Email Logs** [New Email Log]

Action	Email Log Number
<a href="#">Edit</a>   <a href="#">Del</a>	MSG-08695
<a href="#">Edit</a>   <a href="#">Del</a>	MSG-08696

The log file below is a child of the parent we just looked at. Its status message states that there was an error with the parent (the governor limit) that caused this message to fail.

### Email Log Detail

[Edit](#) [Delete](#) [Clone](#) [Resend Message\(s\)](#) [Resend Failed Messages Only](#)

Log Type Summary This is a child log for a single email message. Other emails were sent. To see them, go to the parent email log to get to any child.

Email Log Number	MSG-08695	Owner	Steven Schneider <a href="#">(Change)</a>
Email Setting Name	templateTests	Sent Status	Failed
Target Email Address		Status Message	Parent failure: Exception sending email: The daily limit for the org would be exceeded by this request
Template Name		Target Object Type	
Template Population Object Id		Target Email Object Id	
Template Population Object Type		Parent Email Log	MSG-08694
Source SObject Id String	a01000000WINKbAAN	Has Children	<input type="checkbox"/>
Was Sent	<input type="checkbox"/>		
Created By	Steven Schneider, 4/14/2015 10:52 PM	Last Modified By	Steven Schneider, 4/14/2015 10:52 PM

Log Details [\(Debug\)](#) Email Sender : FCM\_MessageSender : Read String Field ( Id ) returns: a01000000WINKbAAN  
 (Debug) Email Sender : FCM\_MessageSender : Read String Field ( null ) returns: null  
 (Debug) Email Sender : FCM\_MessageSender : Read String Field ( Sender\_Email\_\_c ) returns: synthought@gmail.com  
 (Debug) Email Sender : FCM\_MessageSender : Read String Field ( Sender\_Name\_\_c ) returns: Steve Sender  
 (Debug) Email Sender : FCM\_MessageSender : Read String Field ( null ) returns: null  
 (Debug) Email Sender : FCM\_MessageSender : Read String Field ( Priority\_\_c ) returns: Highest  
 (Debug) Email Sender : FCM\_MessageSender : Read String Field ( Message\_Subject\_\_c ) returns: big attachment  
 (Debug) Email Sender : FCM\_MessageSender : Read String Field ( Template\_ID\_\_c ) returns: First\_Class\_Messagin...  
 (Debug) Email Sender : FCM\_MessageSender : Being sent to email address: null  
 (Debug) Email Sender : FCM\_MessageSender : Read String Field ( Name ) returns: ERN-9395  
 (Debug) Email Sender : FCM\_MessageSender : To Email Address: null toName: ERN-9395  
 (Debug) Email Sender : FCM\_MessageSender : What id (merge data object): a01000000WINKbAAN  
 (Debug) Email Sender : FCM\_MessageSender : Error: There was an attachment that was larger than the max size.

[Edit](#) [Delete](#) [Clone](#) [Resend Message\(s\)](#) [Resend Failed Messages Only](#)

[Chat](#)

The log messages also have buttons to resend a message or just failed messages. When clicked at the parent level, a bulk set of messages can be resent or just failed messages. This can be particularly useful in cases like this where there might not be anything wrong with the messages, themselves, but we have simply hit a governor limit.

## Setting up a Trigger

When **First Class Messaging** is installed, it comes with a trigger on its example class, `Email_Request__c`. You are welcome to use this class, but keep in mind that, as part of a managed package, this trigger cannot be edited or deactivated. The code for the trigger is below.

```
trigger sendEmail on FirstClass__Email_Request__c (after insert) {

    Boolean createIfMissing = true; // If this custom setting doesn't exist, create it.
    Boolean createActive = true; // If created, start it with the trigger turned on.
    Boolean isTriggerActive =
        FirstClass.FCM_SettingsUtils.isTriggerSwitchActive('Email_Request__After_Insert',
            createIfMissing, createActive);
    if (isTriggerActive) {
        FirstClass.FCM_MessageSender ms = new FirstClass.FCM_MessageSender();
        String settingName = ms.getSettingNameForClass('Email_Request__c');
        if (settingName != null) {
            Set<ID> ids = Trigger.newMap.keySet();
            ms.sendEmails(settingName, ids);
        }
    }
}
```

Some of the classes and methods have been exposed for you to create your own triggers. For example, if you wanted to create a trigger on the Contact object to automatically send a message after a Contact is created, you could do it as follows.

```

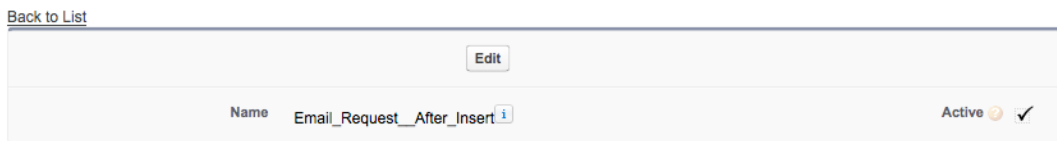
trigger sendEmail on Contact (after insert) {
    Boolean createIfMissing = true; // If this custom setting doesn't exist, create it.
    Boolean createActive = true; // If created, start it with the trigger turned on.
    Boolean isTriggerActive =
        FirstClass.FCM_SettingsUtils.isTriggerSwitchActive('Contact__After_Insert',
            createIfMissing, createActive);
    if (isTriggerActive) {
        FirstClass.FCM_MessageSender ms = new FirstClass.FCM_MessageSender();
        String settingName = ms.getSettingNameForClass('Contact');
        if (settingName != null) {
            Set<ID> ids = Trigger.newMap.keySet();
            ms.sendEmails(settingName, ids);
        }
    }
}

```

### Trigger Switches

In the trigger example above, near the top, there is a check for a trigger switch (a call to `isTriggerSwitchActive ('uniqueName', createIfMissingFlag, createActiveFlag)`). This is a utility that is provided for enabling and disabling triggers. When the trigger is first checked, it will create a custom setting that can later be used to turn on and off the trigger without requiring any deployments or changes to the trigger, itself. The setting is called **FCM Trigger Switches Detail**. A record is added for each “Name” that it is called with. A Name is any string you choose, but you should probably stick to a standard such as an object name followed by the type of trigger it is. For example, in the code above, the switch uses 'Contact\_\_After\_Insert' as the name. Below is a switch in the Custom Settings.

#### FCM Trigger Switches Detail



### Calling from Process Builder or a Flow

In recent Salesforce releases, Salesforce has introduced new ways of automating processes. The latest, Process Builder, allows you to call Apex directly from a simple flow. In the example below, if a new Email Request record is added, the **First Class Messaging** mailer is activated.

It is very easy to setup one of these processes. You select the object you want your process to work off of. In the example below, it is the Email Request object. You set a condition, which can be to always execute the action, and you create an action (shown in the image).

The screenshot displays the Salesforce Process Builder interface for a process named "Email Request Created". The top navigation bar includes "Process Builder - Email Request Created", "Last saved on 7/4/2015 11:35 AM", and navigation links for "Back To Setup" and "Help". Below the navigation bar are buttons for "Expand All", "Collapse All", "View All Processes", "Clone", "View Properties", "Deactivate", and "Read Only".

The main workspace is divided into two sections. On the left is a flowchart starting with a "START" node, followed by an "Email Request" object. A decision diamond labeled "Always evaluate" has a "TRUE" path leading to a box with "IMMEDIATE ACTIONS" (containing a "Send email" action) and "SCHEDULED ACTIONS" (containing a "Set Schedule" action), which then leads to a "STOP" node. The "FALSE" path leads to another decision diamond labeled "Add Criteria", which also has a "TRUE" path leading to a similar action box and a "FALSE" path leading to a "STOP" node.

On the right is the "Call Apex" configuration panel. It includes fields for "Action Name" (set to "Send email") and "Apex Class" (set to "FirstClass\_\_InvocableMessageSender"). Below these is a "Set Apex Variables" section with a table:

Field*	Type*	Value*
Single sObject Id	Reference	{FirstClass__Email_Reques... Q}

There are two options (system classes) that can be used to handle requests. The most common will be "InvocableMessageSender". This takes an object Id for the object that is used to control the email. That is all that is required. The system will find the active setting for that object. You do have the option of specifying a particular setting name, however.

The Classes and arguments are shown below.

Apex Class	Apex Variables
<b>InvocableMessageSender</b>	<ul style="list-style-type: none"> <li>• ID – Id of the object used for messaging</li> <li>• Setting Name - Optional. Will look up first active setting if not supplied</li> </ul>
<b>InvocableBulkMessageSender</b>	<ul style="list-style-type: none"> <li>• ID List – Id Set for bulk group of messages.</li> <li>• Setting Name - Optional. Will look up first active setting if not supplied</li> </ul>

## Adding a Send (Resend) Button

If you setup **First Class Messaging** to send messages driven by an object in your org, you may want to add a send / resend button to a page layout for the object. To do this, first you start by creating a VisualForce page that is the result page. An example follows for the contact object. You can use the code below, only changing the object name in the first line.

```
<apex:page standardController="Contact"
    extensions="FirstClass.FCM_MessageResendControllerExtension"
    action="{!resendMessage}"
    sidebar="true" standardStylesheets="true">

    <link rel='stylesheet' type='text/css'
href="{!URLFOR($Resource.FirstClass__fcmstatics, 'resendstyles.css')}"/>
/>

<br/>
<apex:outputPanel >

    <div style="text-align: right; width:80%;"><h2>Do not resubmit this
page, or messages may be sent again.</h2></div>
<br/>
    <apex:panelGrid width="80%" cellpadding="0" cellspacing="0"
        bgcolor="white" columns="2"
styleClass="topPanelContainer">

        <apex:dataTable value="{!resultMessages}" var="resultMsgStr"
            id="theTable" rowClasses="fcmTblOdd,fcmTblEven"
styleClass="fcmTable"
            cellpadding="5" cellspacing="3">
            <apex:column headerClass="fcmTblTitle">
                <apex:facet name="header">Results From Sending
Messages</apex:facet>
                <apex:outputText value="{!resultMsgStr}"/>
            </apex:column>
        </apex:dataTable>
    </apex:panelGrid>
</apex:outputPanel>

</apex:page>
```

Once the result page has been created, you need to create a new button. In this case, you go to **Customize > Contacts > Buttons, Links, and Actions** . This feature can be found at the top of the page for custom objects.

When you create the button or link as seen below, the name of the VisualForce page that you just created will automatically be in the Content dropdown list.

The button is then available to place on page layouts in the Custom Buttons section.

Edit Contact Custom Button or Link  
**Resend Message**

**Custom Button or Link Edit** Save Quick Save Preview Cancel

Label

Name  i

Description

Display Type  
 Detail Page Link [View example](#)  
 Detail Page Button [View example](#)  
 List Button [View example](#)

Behavior  View Behavior Options

Content Source

**Content**

Save Quick Save Preview Cancel

## Governor Limits

**First Class Messaging** enhances Salesforce’s emailing capabilities, but it is still governed by the limits of Salesforce. For example, when Salesforce’s SingleEmailMessage Apex call is used, the following limits are in place (as of Summer 2015 Salesforce release). Note that these limits are subject to change in future Salesforce releases, and it is also possible to request to have limits raised by Salesforce.

\* Source is [Salesforce Limits Quick Reference Guide](#), Version 34, Summer 2015

*Using the API or Apex, you can send single emails to a maximum of 1,000 external email addresses per day based on Greenwich Mean Time (GMT). Single emails sent using the Salesforce application don't count toward this limit. There's no limit on sending individual emails to contacts, leads, person accounts, and users in your organization directly from account, contact, lead, opportunity, case, campaign, or custom object pages.*

- *You can send 100 emails per SingleEmailMessage.*
- *If you use SingleEmailMessage to email your organization’s internal users, specifying the user’s ID in setTargetObjectId means the email doesn’t count toward the daily limit. However, specifying internal users’ email addresses in setToAddresses means the email does count toward the limit.*



So, what does this mean?

**First Class Messaging** makes use of SingleEmailMessage, so it is affected by these limits. You can send a message that has up to 100 recipients via the CC, BCC, To, and / or “Template Target” fields. Typically, not many recipients will be used for each message.

If you use the “Template Target Field API Name” field in your settings and point to a field with a User Id in it, the messages sent to those users should not count against your daily limits.

So, what happens if you hit the limit?

**First Class Messaging** will log the error and can send an alert message to an administrative user or to a Chatter group (if the appropriate settings are populated).

In addition the limits guide states:

Maximum Size of Email Message (includes email headers, body, attachments, and encoding): 25 MB for Professional Edition, Enterprise Edition, Unlimited and Performance Edition, Developer Edition.

This means that when an email is created that has attachments included, the total size of the email body plus attachments cannot exceed 25 MB. **First Class Messaging** keeps track of the sizes but also watches the memory. If you try to send a large bulk group of emails (i.e., via data loader upload or by selecting a number of items and clicking [send]), you may get a message about having hit another governor limit – heap memory.

Basically, **First Class Messaging** does its best at delivering all kinds of emails and bulk sets of messages, but running in a multitenant environment, it is still governed by Salesforce’s governor limits. It is best to try to limit the size of attachments and the number of bulk messages when using attachments, whenever possible. Of course, if you receive an error sending a message, you can always adjust it by removing an attachment or sending messages individually by using a send or resend button.

## Validation

Once the “Emailing Active” setting is checked, a number of validations are made. This is part of the power of **First Class Messaging**. If you were to code an emailing routine by hand, you would need to take into account combinations of settings. For example, if you use an email template, there are a number of other fields that are required, such as a Template Target Object (i.e., a lookup Id for a Contact). If this is supplied, you really don’t need a Recipient Name or Recipient Email provided. There

are minimum fields needed when sending messages with and without templates and with and without calendar attachments. **First Class Messaging** catches many of these errors when the settings are created. If they can't be caught ahead of time (for example, if there are fields on the object that are needed but are not entered when the email object is created, these are caught when the message is created and are reported back in logs and in alerts.

## Troubleshooting

Error Message	Description and Resolution
Error executing query: No such column 'missingFieldName__c' on entity 'SomeObject__c'.	<p>The name of a field in the settings no longer exists in the Object (table). If the field was removed or renamed, this will happen.</p> <p>To Resolve, modify the active settings for the Object class in the message and look for the reference to the old field. Edit and select a new field from the drop-down.</p>
System.EmailException: SendEmail failed. First exception on row 0; first error: NO_MASS_MAIL_PERMISSION, Single email is not enabled for your organization	<p>The org does not have access to send email turned on.</p> <p>Go to Setup &gt; Email Administration &gt; Deliverability Change "Access to Send Email" Access level to "All email".</p>
Error sending (status UNVERIFIED_SENDER_ADDRESS), (target null) :Organization-Wide Email Address has not be verified for use. (message is in log file)	<p>This confusingly written error message is from the system. It means that you are using an "Org-Wide Sender Email Address" that has not been successfully verified. A message is sent upon setup that must be accepted to ensure valid email addresses.</p>

Please send any feedback to us at [support@beanstalkcloud.com](mailto:support@beanstalkcloud.com) so that we can improve the troubleshooting guide, the documentation, and the product, itself.

Thank you very much for selecting our product.