

IccEmailer Manual

Contents

Description.....	2
IccEmailer.....	2
Installation.....	2
Using.....	3
Starting	3
Starting with an alternate Logic File.....	3
Logic File Description/Syntax.....	3
Logic File – Category: Logging.....	4
Logic File – Category: Decryption	5
Logic File – Category: SMTP Server/Connections.....	6
Logic File – Email Packets.....	7
Logic File – Replace Tags	13
Logic File – Message Personalization.....	14
Logic File – Example	15
Encryption.....	18
Definitions	19
Modifications.....	19

Description

This document describes how to use lccEmailer.

lccEmailer

is a Command Line program created to send emails, multiple capabilities, including mail merging, attachments, etc.

The program is controlled by a Logic File, from one to many different ones.

Please see the distribution site/page for example Logic Files.

Capabilities include:

- specify format of message, i.e. Plain Text, or HTML
- loading sender information from the Logic File or an external file
- sending from one to many recipients
- sending personal parts of the message only to specific recipients
- sending attachments, both to all and to specific recipients
- loading recipients from the Logic File and/or external files (lists)
- loading the subject from the Logic File and/or external file
- loading messages from the Logic File and/or external files (lists)
- mail merge/replace content in Subject and/or message for Sender/Recipient information
- supply multiple messages per Logic File
- encryption of sensitive values in the Logic File

Installation

- copy the lccEmailer.exe to a folder
- create a Logic File, depending on the need
- run the lccEmailer.exe

Using

Prerequisites: configure a Logic File.

The program will default to using a Logic File called:

`lccEmailer-logic.txt`

in the same folder as the program. You can change which Logic File to use by specifying a different one on the command line.

Starting

- run the `lccEmailer.exe`

Starting with an alternate Logic File

- run the `lccEmailer.exe lcc:logicPath [Logic File]`
- ex: `lccEmailer.exe lcc:logicPath ourLogic.txt`

Logic File Description/Syntax

The Logic File is a Tab delimited text file. Any lines not recognized as a valid Key/Value pair, will be ignore and can be used as remarks/other.

The Logic File uses the syntax.

Syntax: `[Key] [tab] [Value]`

Example: `lcc:key value`

Most Keys/Values only use two columns, the 1st column for the Key, the 2nd column for the value. Some Keys have multiple columns as needed. This is a nice way to document specific Key settings (see Log Levels in the Logic File example(s) for reference). Also, if you place a tab before a line, that will essentially make it a remark and will be ignored, which makes using/not using logic without removing quicker.,

Though you can place the Key/Values in any order, the program will read/execute the settings/actions as they are loaded, per the category of key.

There are categories, as they are executed:

- Logging
- Encryption Keys (optional)
- SMTP Server/Connection Keys
- Email Packets

To make creating/maintaining your logic file easier, we recommend placing your keys in the same order. i.e. place your Logging keys at top, followed by Encryption Keys (optional), SMTP Server/Connection Keys, then finally the Email Packets.

All keys can have encrypted values. To specify that a key value is encrypted, prepend the value with "[lcc:encrypted]":

If you supply encrypted keys, you must also provide these keys:

- lcc:sourceEncryptedPasswordHash
- lcc:sourceEncryptedSaltKey
- lcc:sourceEncryptedVIKey

Logic File – Category: Logging

The Logging category instructs the program what to give feedback on. Most feedback is logged to a log file, but, some basic information is also copied to the screen.

The first setting in your Logic File should be to determine what your log file path/name will be.

lcc:logPath *(optional, but highly recommended)(one per Logic File)*

The path/root name of the Log File. The program will place the '.log' extension automatically and will also append a Year/Month date.

Example: **lcc:logPath lccEmailerLog**

lcc:logLevel *(optional)(none to many per Logic File)*

Specifies what log information will be logged. Provide a new line for each level desired.

Log Levels

- 1 **Load Logic:** display loading the Logic File (available on the command line only)
- 2 **Check Email Packet Id Unique:** displays checks performed on the Keys supplied by the Logic File.
- 3 **Email Packet Settings:** display information about logic being processed.

- 4 Load Logic Keys/Values: display the Keys/values being loaded (available on the command line only).
- 5 Process Email Packets: the File Ids loaded.
- 6 Send Email: file settings loaded from the Logic File.
- 7 Functions: display loading source files.
- 8 Encryption: display loading a source file.

Syntax: `lcc:logLevel [tab] [#]`

Example: `lcc:logLevel 3`

Example: `lcc:logLevel 3`

`lcc:logLevel 5`

Logic File – Category: Decryption

The Decryption category instructs the program whether to use decryption and what Keys to use to perform it. See Encryption section for instructions on encrypting values.

lcc:sourceEncryptedPasswordHash *(optional)(one per Logic File)*

Hash value supplied for encryption. Must be used with keys 'lcc:sourceEncryptedSaltKey' and 'lcc:sourceEncryptedVIKey'.

Any set of characters can be used.

Syntax: `lcc: sourceEncryptedPasswordHash [tab] [...]`

Example: `lcc: sourceEncryptedPasswordHash s0m3C00lPhr@$e`

lcc:sourceEncryptedSaltKey *(optional)(one per Logic File)*

Salt Key value supplied for encryption. Must be used with keys 'lcc:sourceEncryptedPasswordHash' and 'lcc:sourceEncryptedVIKey'.

Any set of characters can be used.

Syntax: `lcc: sourceEncryptedSaltKey [tab] [...]`

Example: `lcc: sourceEncryptedSaltKey s0m3C00lPhr@$e`

lcc:sourceEncryptedVIKey *(optional)(one per Logic File)*

VI Key value supplied for encryption. Must be used with keys 'lcc:sourceEncryptedPasswordHash' and 'lcc:sourceEncryptedSaltKey'.

Any set of characters can be used, but, must be exactly 16 characters long.

Syntax: **lcc:sourceEncryptedVIKey** *[tab]* **[...]**

Example: **lcc:sourceEncryptedVIKey** s0m3C00lPhr@\$e12

Logic File – Category: SMTP Server/Connections

The SMTP Server/Connection category instructs the program how to connect to the email server.

If the User and Password are not supplied, connection is assumed to be anonymous.

lcc:SMTPServer *(mandatory)(one per Logic File)*

What is the server address.

Syntax: **lcc:SMTPServer** *[tab]* **[address]**

Example: **lcc:SMTPServer** ouremail.ourcollege.edu

lcc:SMTPPort *(mandatory)(one per Logic File)*

What is the serverport.

Syntax: **lcc:SMTPPort** *[tab]* **[#]**

Example: **lcc:SMTPPort** 25

lcc:SMTPSSL *(optional)(one per Logic File)*

Enables SSL. Currently only TLS (Transport Layer Security) is supported, SMTP over SSL method is not.

Valid Values

- YES

Syntax: **lcc:SMTPSSL** *[tab]* **[YES]**

Example: **lcc:SMTPSSL** YES

lcc:SMTPUser *(optional)(one per Logic File)*

What is the User Id to connect with.

We highly recommend using the Encryption section to encrypt this value.

Syntax: `lcc:SMTPUser [tab] [id]`

Example: `lcc:SMTPUser jdoe@ourcollege.edu`

Example (encrypted): `lcc:SMTPUser [lcc:encrypted]A1B2C3==`

lcc:SMTPPassword *(optional)(one per Logic File)*

What is the User Password to connect with.

We highly recommend using the Encryption section to encrypt this value.

Syntax: `lcc:SMTPPassword [tab] [id]`

Example: `lcc:SMTPPassword some cool password`

Example (encrypted): `lcc:SMTPPassword [lcc:encrypted]A1B2C3==`

lcc: SMTPSendDelay *(optional)(one per Logic File)*

By default, the program will delay 1000 milliseconds (1 second) between each email sent (recipient). This key will change that default.

The value should be in milliseconds.

Syntax: `lcc:SMTPSendDelay [tab] [#]`

Example: `lcc:SMTPSendDelay 5000`

`lcc:SMTPSendDelay`

Logic File – Email Packets

A Logic File can contain one to many Email Packets. Each Packet is considered a separate message that is being sent to recipients.

Each Packet should always start with the key "lcc:emailId". When the program comes across a logic line with that Key, it assumes a new Packet is being configured. Any Keys after this Key, but, before the next "lcc:emailId" is considered Keys for the current Packet.

The mandatory Keys are:

- lcc:emailId
- lcc:emailSender
 - or- lcc:emailSenderFromFile
- lcc:emailRecipient
 - or- lcc:emailRecipientFromFile
 - or- lcc:emailRecipientCC
 - or- lcc:emailRecipientCCFromFile
 - or- lcc:emailRecipientBCC
 - or- lcc:emailRecipientBCCFromFile
- lcc:emailSubject
 - or- lcc:emailSubjectFromFile
- lcc:emailMessage
 - or- lcc:emailMessageFromFile

All other Keys are optional.

lcc:emailId *(mandatory)(one per Email Packet)*

The Id of the current Email Packet. Each Id must be unique in the same Logic File.

Syntax: **lcc:emailId** [tab] [value]

Example: **lcc:emailId** our cool email packet

lcc:emailSkip *(optional)(one per Email Packet)*

Should be placed right after the "lcc:emailId" Key. If provided and the value is "YES", this packet will not be processed. This is a good way to turn on/off packets, without removing from the Logic File.

Valid Values

- YES

Syntax: **lcc:emailSkip** [tab] [YES]

Example: **lcc:emailSkip** YES

lcc:emailFormat *(optional)(one per Email Packet)*

Provides the Format of the message. Messages default to plain text, unless "HTML" is supplied. If supplied, the message is formatted to display as a web page.

Valid Values

- HTML

Syntax: **lcc:emailFormat** [tab] [HTML]

Example: **lcc:emailFormat** HTML

lcc:emailSender *(mandatory or use "lcc:emailSenderFromFile")(one per Email Packet)*

Provides the sender Email Address, and optional Name.

Syntax: **lcc:emailSender** [tab] [Address] [tab] [Name]

Example: **lcc:emailSender** ourperson@ourcollege.edu Our Person's Name

lcc:emailSenderFromFile *(mandatory or use "lcc:emailSender")(one per Email Packet)*

Provides the sender Email Address, and optional Name from a file. Only one should be provided.

The file should be formatted like:

[Address] [tab] [Name]

Syntax: **lcc:emailSenderFromFile** [tab] [Path]

Example: **lcc:emailSenderFromFile** ourSender.txt

Example: **lcc:emailSenderFromFile** \\server\share\$\ourSender.txt

lcc:emailRecipient *(mandatory or use alternative)(one to many per Email Packet)*

Provides the recipient Email Address, and optional Name. This email address will be placed on the TO line.

Syntax: **lcc:emailRecipient** [tab] [Address] [tab] [Name]

Example: **lcc:emailRecipient** ourperson@ourcollege.edu Our Person's Name

Example (multiple): **lcc:emailRecipient** ourperson@ourcollege.edu Our Person's Name
lcc:emailRecipient anotherhuman@some.com Another Human

lcc:emailRecipientCC *(mandatory or use alternative)(one to many per Email Packet)*

Provides the recipient CC Email Address, and optional Name. This email address will be placed on the CC line.

Syntax: `lcc:emailRecipientCC [tab] [Address] [tab] [Name]`

Example: `lcc:emailRecipientCC ourperson@ourcollege.edu Our Person's Name`

Example (multiple): `lcc:emailRecipientCC ourperson@ourcollege.edu Our Person's Name`
`lcc:emailRecipientCC anotherhuman@some.com Another Human`

lcc:emailRecipientBCC *(mandatory or use alternative)(one to many per Email Packet)*

Provides the recipient BCC Email Address, and optional Name. This email address will be placed on the BCC line.

Syntax: `lcc:emailRecipientBCC [tab] [Address] [tab] [Name]`

Example: `lcc:emailRecipientBCC ourperson@ourcollege.edu Our Person's Name`

Example (multiple): `lcc:emailRecipientBCC ourperson@ourcollege.edu Our Person's Name`
`lcc:emailRecipientBCC anotherhuman@some.com Another Human`

lcc:emailRecipientToEmpty *(optional)(one per Email Packet)*

If CC or BCC key options used on recipient, this key allows you to fill-in the TO box with an address. Since the program sends out an individual email to each recipient, CC and BCC emails would have a blank TO line.

Syntax: `lcc:emailRecipientToEmpty [tab] [Address] [tab] [Name]`

Example: `lcc:emailRecipientToEmpty ourperson@ourcollege.edu Our Person's Name`

Example (multiple): `lcc:emailRecipientToEmpty ourperson@ourcollege.edu Our Person's Name`
`lcc:emailRecipientToEmpty anotherhuman@some.com Another Human`

lcc:emailRecipientFromFile *(mandatory or use an alternative)(one to many per Email Packet)*

Provides the recipient Email Address, and optional Name from a file. One to many can be provided. These email addresses will be placed on the TO line.

The file should be formatted like:

`[Address] [tab] [Name]`

...

`[Address] [tab] [Name]`

Syntax: `lcc:emailRecipientFromFile [tab] [Path]`

Example: `lcc:emailRecipientFromFile ourSender.txt`

Example (multiple): **lcc:emailRecipientFromFile** ourRecipients.txt
lcc:emailRecipientFromFile \\server\share\$\ ourRecipients.txt

lcc:emailRecipientCCFromFile *(mandatory or use an alternative)(one to many per Email Packet)*

Provides the recipient CC Email Address, and optional Name from a file. One to many can be provided. These email addresses will be placed on the CC line.

The file should be formatted like:

```
[Address] [tab] [Name]  
...  
[Address] [tab] [Name]
```

Syntax: **lcc:emailRecipientCCFromFile** [tab] [Path]

Example: **lcc:emailRecipientCCFromFile** ourSender.txt

Example (multiple): **lcc:emailRecipientCCFromFile** ourRecipients.txt
lcc:emailRecipientCCFromFile \\server\share\$\ ourRecipients.txt

lcc:emailRecipientBCCFromFile *(mandatory or use an alternative)(one to many per Email Packet)*

Provides the recipient BCC Email Address, and optional Name from a file. One to many can be provided. These email addresses will be placed on the BCC line.

The file should be formatted like:

```
[Address] [tab] [Name]  
...  
[Address] [tab] [Name]
```

Syntax: **lcc:emailRecipientBCCFromFile** [tab] [Path]

Example: **lcc:emailRecipientBCCFromFile** ourSender.txt

Example (multiple): **lcc:emailRecipientBCCFromFile** ourRecipients.txt
lcc:emailRecipientBCCFromFile \\server\share\$\ourRecipients.txt

lcc:emailSubject *(mandatory)(one per Email Packet)*

Provides the subject line.

See the Replace Tags section for auto changing subject contents.

Syntax: `lcc:emailSubject [tab] [subject]`
Example: `lcc:emailSubject our cool subject`

lcc:emailMessage (mandatory or use "lcc:emailMessageFromFile")(one to many per Email Packet)
Provides the message.

See the Replace Tags section for auto changing message contents.

See the key "lcc:emailMessageFromFile" and the Message Personalization section for filtering what lines are sent to who.

If you have the Format of the message as HTML, make sure to format your information in HTML.

Syntax: `lcc:emailMessage [tab] [message]`
Example: `lcc:emailMessage our cool message`
Example (multiple): `lcc:emailMessage our cool message line #1`
`lcc:emailMessage our cool message line #2`

lcc:emailMessageFromFile (mandatory or use "lcc:emailMessage")(one to many per Email Packet)
Provides the message from a file.

See the Replace Tags section for auto changing message contents.

See the Message Personalization section for filtering what lines are sent to who.

If you have the Format of the message as HTML, make sure to format your information in HTML.

The file should be formatted like:

`[message line...]`
`[message line...]`

Syntax: `lcc:emailMessageFromFile [tab] [Path]`
Example: `lcc:emailMessageFromFile ourMessage.txt`
Example (multiple): `lcc:emailMessageFromFile ourMessage.txt`
`lcc:emailMessageFromFile \\server\share$\ourMessage.txt`

lcc:emailAttachment *(optional)(one to many per Email Packet)*

Provides a file attachment.

Syntax: **lcc:emailAttachment** [tab] [path]

Example: **lcc:emailAttachment** ourFile.pdf

Example (multiple): **lcc:emailAttachment** ourFile.pdf
lcc:emailAttachment \\server\share\$\ourFile.txt

lcc:emailPersonalAttachment *(optional)(one to many per Email Packet)*

Provides a file attachment, but, only to specific recipients.

If more than one recipient is specified, separate the address by commas, with no spaces.

Syntax: **lcc:emailPersonalAttachment** [tab] [recipients] [tab] [path]

Example: **lcc:emailPersonalAttachment** ourperson@ourcollege.edu ourFile.pdf

Example (multiple): **lcc:emailPersonalAttachment** ourperson@ourcollege.edu,jdoe@doej.com ourFile.txt
lcc:emailPersonalAttachment ourperson@ourcollege.edu ourFile.pdf

Logic File – Replace Tags

The program will replace the following tags with the associated value in the following sections:

- subject
- message

[lcc:senderAddress] *(none to many per lines)*

Replaces with the sender's address. Place the Tag in any place you want the value to be placed.

Syntax: **[lcc:senderAddress]**

Example (before change): a cool sentence for **[lcc:senderAddress]** has been provided.

Example (after change): a cool sentence for **ourperson@ourcollege.edu** has been provided.

[lcc:senderName] *(none to many per lines)*

Replaces with the sender's name. Place the Tag in any place you want the value to be placed.

Syntax: **[lcc:senderName]**

Example (before change): a cool sentence for [lcc:senderName] has been provided.

Example (after change): a cool sentence for John Doe has been provided.

[lcc:subject] *(none to many per lines)*

Replaces with the subject line. Place the Tag in any place you want the value to be placed.

Syntax: [lcc:subject]

Example (before change): we sent you a message with subject [lcc:subject].

Example (after change): we sent you a message with subject Welcome Home.

[lcc:recipientAddress] *(none to many per lines)*

Replaces with the recipient's address. Place the Tag in any place you want the value to be placed.

Syntax: [lcc:recipientAddress]

Example (before change): a cool sentence for [lcc:recipientAddress] has been provided.

Example (after change): a cool sentence for ourperson@ourcollege.edu has been provided.

[lcc:recipientName] *(none to many per lines)*

Replaces with the recipient's name. Place the Tag in any place you want the value to be placed.

Syntax: [lcc:recipientName]

Example (before change): a cool sentence for [lcc:recipientName] has been provided.

Example (after change): a cool sentence for John Doe has been provided.

Logic File – Message Personalization

The program will filter lines from a Message File (see Key "lcc:emailMessageFromFile") depending on the recipient's address.

Start any line you want to filter in the message file with any of the Keys below.

[lcc:personalMessage] *(one per message line)*

If specified at the beginning of a message line, the program will look at the 2nd column for recipients who are authorized to receive the line. Finally the last column is the line to be provided.

If you have the Format of the message as HTML, make sure to format your information in HTML.

Syntax: [lcc:personalMessage] [tab] [recipients] [tab] [message]

Example: [lcc:personalMessage] jdoe@somesite,jane@another.org cool sentence only for Joe and Jane

[lcc:personalMessageFromFile] *(one per message line)*

If specified at the beginning of a message line, the program will look at the 2nd column for recipients who are authorized to receive this information. Finally the last column is the file contents to be provided.

The contents of the file being provided will be provided as-is.

If you have the Format of the message as HTML, make sure to format your information in HTML.

Syntax: [lcc:personalMessageFromFile] [tab] [recipients] [tab] [message]

Example: [lcc:personalMessageFromFile] jdoe@somesite,jane@another.org ourFile.txt

Logic File – Example

This logic file shows multiple email template. Since you can one to many templates (starting with 'lcc:emailId'), and you can turn the 'skip' option on/off on each template.

lcc:logPath lccEmailerLog

```
lcc:logLevel 1 Load Logic
lcc:logLevel 2 check Email Id unique
lcc:logLevel 3 Email Packet Settings
lcc:logLevel 4 Load Logic Keys/Values
lcc:logLevel 5 Process Email Packets
lcc:logLevel 6 Send Email
lcc:logLevel 7 Functions
lcc:logLevel 8 Encryption
```

lcc:sourceEncryptedPasswordHash...

lcc:sourceEncryptedSaltKey...

lcc:sourceEncryptedVIKey ...

lcc:SMTPServer ouremailserver.ourcollege.edu
lcc:SMTPPort 25

lcc:SMTPUser [lcc:encrypted]...
lcc:SMTPPassword [lcc:encrypted]...

lcc:emailId Test Email #1, Basic 'Hello Word' Example
lcc:emailSkip YES
lcc:emailSender oursender@ourcollege.edu David Mielcarek
lcc:emailRecipient arecipient@ourcipientemailserverdomain.com David Mielcarek @ My.LCC
lcc:emailRecipient arecipient@ourcipientemailserverdomain.com David Mielcarek @ LCC
lcc:emailSubject Test Subject #1, Basic 'Hello Word' Example
lcc:emailMessage Test Message, "Hello World"

lcc:emailId Test Email #2, Basic 'Hello Word' Example, in HTML
lcc:emailSkip YES
lcc:emailFormat HTML
lcc:emailSender oursender@ourcollege.edu David Mielcarek
lcc:emailRecipient arecipient2@anotheremailserverdomain.com David Mielcarek
lcc:emailSubject Test Subject #2, Basic 'Hello Word' Example, in HTML
lcc:emailMessage <html>
lcc:emailMessage <head>
lcc:emailMessage <title>Test Email #2, Basic 'Hello Word' Example, in HTML</title>
lcc:emailMessage </head>
lcc:emailMessage <body>
lcc:emailMessage Test Message, "Hello
World";
lcc:emailMessage </body>
lcc:emailMessage </html>

lcc:emailId Test Email #3, Basic 'Hello Word' Example, with attachment
lcc:emailSkip YES
lcc:emailSender oursender@ourcollege.edu David Mielcarek
lcc:emailRecipient arecipient2@anotheremailserverdomain.com David Mielcarek

lcc:emailSubject Test Subject #3, Basic 'Hello Word' Example, with attachment
lcc:emailMessage Test Message, "Hello World"
lcc:emailAttachment lccEmailerTestAttachment.pdf

lcc:emailId Test Email #4, Recipient/Subject/Message From File
lcc:emailSkip YES
lcc:emailSenderFromFile sender2.txt
lcc:emailRecipientFromFile lccEmailerTestRecipients.txt
lcc:emailSubjectFromFile lccEmailerTestSubject.txt
lcc:emailMessage Test line from Logic File - Line 1
lcc:emailMessageFromFile lccEmailerTestMessage.txt
lcc:emailMessage Test line from Logic File - Line 2

lcc:emailId Test Email #5, in HTML, Recipient/Subject/Message From File
lcc:emailSkip YES
lcc:emailFormat HTML
lcc:emailSenderFromFile lccEmailerTestSender.txt
lcc:emailRecipientFromFile lccEmailerTestRecipients.txt
lcc:emailSubjectFromFile lccEmailerTestSubject2.txt
lcc:emailMessage <html><head><title>Test Email #5, in HTML, Recipient/Subject/Message From File</title></head>
lcc:emailMessageFromFile lccEmailerTestMessage2.txt

lcc:emailId Test Email #6, in HTML, personal messages (per email)
lcc:emailSkip YES
lcc:emailFormat HTML
lcc:emailSenderFromFile lccEmailerTestSender.txt
lcc:emailRecipientFromFile lccEmailerTestRecipients.txt
lcc:emailSubjectFromFile lccEmailerTestSubject3.txt
lcc:emailMessageFromFile lccEmailerTestMessage3.txt

lcc:emailId Test Email #7, in HTML, personal messages and personal files (per email)
lcc:emailSkip YES
lcc:emailFormat HTML
lcc:emailSenderFromFile lccEmailerTestSender.txt
lcc:emailRecipientFromFile lccEmailerTestRecipients.txt
lcc:emailSubjectFromFile lccEmailerTestSubject4.txt

lcc:emailMessageFromFile lccEmailerTestMessage4.txt

lcc:emailId Test Email #8, with attachment, with personal attachment

lcc:emailSkip YES

lcc:emailSender oursender@ourcollege.edu David Mielcarek

lcc:emailRecipient arecipient2@anotheremailserverdomain.com David Mielcarek

lcc:emailRecipient arecipient@ourcipientemailserverdomain.com David Mielcarek

lcc:emailSubject Test Email #8, with attachment, with personal attachment

lcc:emailMessage Test Email #8, with attachment, with personal attachment

lcc:emailAttachment lccEmailerTestAttachment.pdf

lcc:emailPersonalAttachment arecipient2@anotheremailserverdomain.com lccEmailerTestAttachment.txt

lcc:emailPersonalAttachment

arecipient2@anotheremailserverdomain.com,arecipient@ourcipientemailserverdomain.com

lccEmailerTestAttachment2.txt

lcc:emailId Notification - Account Deletion

lcc:emailSkip YES

lcc:emailSender our.notifications@ourcollege.edu Account Notifications

lcc:emailRecipientFromFile accountDeletions-list.txt

lcc:emailRecipientCC arecipient@ourcipientemailserverdomain.com

lcc:emailRecipient arecipient2@anotheremailserverdomain.com

lcc:emailRecipientCC arecipient2@anotheremailserverdomain.com

lcc:emailRecipientBCC arecipient2@anotheremailserverdomain.com

lcc:emailRecipientCCFromFile recipients-CC-list.txt

lcc:emailRecipientBCCFromFile recipients-BCC-list.txt

lcc:emailRecipientToEmpty our.notifications@ourcollege.edu

lcc:emailSubject Notification - Account Deletion

lcc:emailMessageFromFile accountDeletion-message.txt

Encryption

To encrypt a value, run the following from a Command Line:

lccEmailer.exe lcc:encryptValue "..."

Syntax: `lccEmailer.exe lcc:encryptValue "[value]"`
Example: `lccEmailer.exe lcc:encryptValue "our cool value"`

Definitions

SMTP - Simple Mail Transfer Protocol

Modifications

NAME	DATE	MODIFICATION
David Mielcarek	2/26/2016	Created
David Mielcarek	2/29/2016	Added Key lcc:SMTPSSL
David Mielcarek	8/3/2017	Added keys 'lcc:emailRecipientCC', 'lcc:emailRecipientBCC', 'lcc:emailRecipientCCFromFile', 'lcc:emailRecipientBCCFromFile', 'lcc:emailRecipientToEmpty'
David Mielcarek	1/26/2024	Added key 'lcc:SMTPSendDelay'

End of document