ColdFusion Foundations: SMTP

Mosh Teitelbaum
mosh.teitelbaum@evoch.com
evoch, LLC

SMTP: Simple Mail Transfer Protocol

- Purpose

  To transfer mail reliably and efficiently *- RFC 2821*

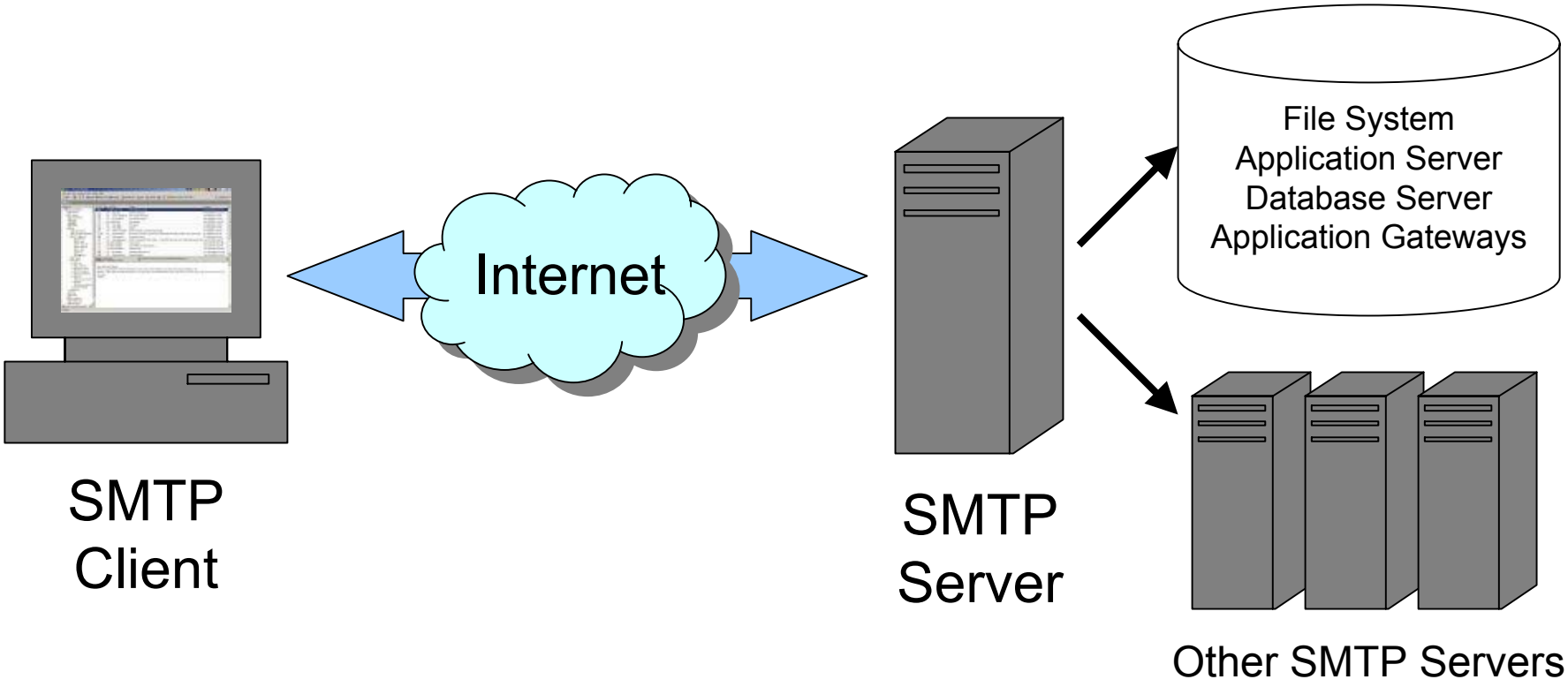
- What is SMTP?

  The protocol that defines how email clients communicate with email servers to send email messages.

- What does SMTP do?

  It allows email messages to be sent and email addresses to be verified. It does not support receipt of email messages by email clients.

SMTP Involves Clients and Servers



SMTP Communication Process

1. SMTP Client determines IP Address of SMTP server by resolving destination domain name to intermediate Mail eXchanger host or final target host.
2. Session Initiation - Client establishes 2-way connection to server (port 25) which responds with welcome message
3. Client Initiation - Client sends identification and server responds with another welcome message
4. Mail Transaction(s) - Mail Objects are transported
5. Termination - Client initiates termination of connection and server replies and terminates the connection

SMTP Commands and Replies

All client-server communication involves:

- Commands
  - Clients send commands to provide information and instructions to the server
- Replies
  - Servers respond with numeric Reply Codes to inform the client of the outcome of the command
  - Replies also include text which is non-standardized and is more for people than for software

SMTP Commands

Command	Description
HELO, EHLO	Identifies the client to the server
MAIL	Initiates a mail transaction. Includes sender's email address
RCPT	Specifies an individual recipient's email address. May be used multiple times to specify multiple recipients.
DATA	Initiates transfer of mail contents
RSET	Resets/aborts the current mail transaction
VRFY, EXPN	Verifies the authenticity of an email address
HELP	Asks the server to send helpful information to the client
NOOP	No operation. Do nothing
QUIT	Instructs the server to terminate the connection

SMTP Reply Code Structure

Reply codes consist of 3 digits: *xyz*

- *x* – Denotes whether the response is good, bad, or incomplete
- *y* – Specifies the type of error (syntax, information, connections, mail system)
- *z* – Provides a specific reason for why the specified type of error occurred

Reply codes followed by a dash (*xyz*-) indicate the presence of one or more additional lines

SMTP Commonly Used Replies

Reply Code	Description
220	Sent by the server upon acceptance of the initial client connection.
250	Sent after successful completion of a command
354	Send after acceptance of the DATA command to instruct the client to send the email content
221	Sent by the server before it terminates the connection

SMTP Session Initiation

After the client creates a connection, the server responds with a welcome message:

```
220 mail805.megamailservers.com ESMTP Sendmail
➡8.12.10/8.12.9; Thu, 6 May 2004 12:34:08 -0400
```

SMTP Client Initiation

After the client receives the welcome message, it identifies itself to the server:

```
220 smtp.evoch.com ESMTP Sendmail 8.12.10/8.12.9;
➡ Thu, 6 May 2004 12:34:08 -0400
EHLO mosh.evoch.com
250-smtp.evoch.com Hello mosh.evoch.com [192.168.1.100],
➡ pleased to meet you
250-ENHANCEDSTATUSCODES
250-PIPELINING
250-8BITMIME
250-SIZE 52428800
250-DSN
250-ETRN
250-AUTH PLAIN LOGIN
250-DELIVERBY
250 HELP
```

SMTP Termination

After the Mail Objects have been sent, the client initiates connection termination:

```
QUIT
221 2.0.0 smtp.evoch.com closing connection
```

SMTP Mail Objects

SMTP transports Mail Objects which contain:

- Envelope
  - Originator Address
  - Recipient Address(es)
  - Optional protocol extension material
- Content
  - Headers
  - Body

SMTP Mail Envelope

Mail Envelopes specify a sender address and one or more recipient addresses

```
MAIL FROM: <mosh@evoch.com>
250 2.1.0 <mosh@evoch.com>... Sender ok
RCPT TO: <someGuy@evoch.com>
250 2.1.5 <someGuy@evoch.com>... Recipient ok
RCPT TO: <someGuy2@evoch.com>
250 2.1.5 <someGuy2@evoch.com>... Recipient ok
RCPT TO: <someGuy3@evoch.com>
250 2.1.5 <someGuy3@evoch.com>... Recipient ok
```

SMTP Mail Contents

Mail Contents consist of headers and a body

```
DATA
354 Enter mail, end with "." on a line by itself
From: "Mosh Teitelbaum" <mosh@evoch.com>
To: "Mosh Teitelbaum" <mosh@evoch.com>
Cc: someGuy@evoch.com, someGuy2@evoch.com, someGuy3@evoch.com
Subject: This is an SMTP Mail Transaction
Date: Thu, 6 May 2004 11:01:12 -0700
[... More headers ...]

The body goes here. The body is ended by
a <CRLF>.<CRLF> sequence - a period on a line by itself.
.
250 2.0.0 i46GY8Du022090 Message accepted for delivery
```

SMTP Common Headers

Some of the more common headers are:

Header	Description
From	Specifies the author(s) email address(es) and display name(s)
To, Cc, Bcc	Specifies the recipient(s) email address(es) and display name(s)
Subject	Specifies the topic of the message
Date	Specifies the date on which the message was originally sent
Reply-To	Specifies the email address to which replies should be sent
Message-ID	Specifies a unique identifier for the message

SMTP Multipart Messages

SMTP supports messages with multiple parts in the Content Body

- Attachments
- Multiple copies of the same email but in different Mime-Types (ex. plain text and HTML)
- Multiple copies of the same email but in different languages

SMTP Multipart Message Structure

- Content-Type header is set to “multipart/mixed” and specifies a unique boundary value

```
Content-Type: multipart/mixed;
 boundary="----=_NextPart_000_018D_01C43446.5F3943C0"
```

- Each part begins and ends with this boundary
- Each part includes its own unique headers related to the content-type, encoding, and disposition of the part

```
Content-Type: application/zip; name="attach.zip"
Content-Transfer-Encoding: base64
Content-Disposition: attachment;
 filename="attach.zip"
```

SMTP Multipart Message Sample

```
Content-Type: multipart/mixed; boundary="----=_NextPart_000_abcdef"

This is a multi-part message in MIME format.

----=_NextPart_000_abcdef
Content-Type: text/plain; charset="iso-8859-1"
Content-Transfer-Encoding: 7bit

This is my message.

----=_NextPart_000_abcdef
Content-Type: application/zip; name="attach.zip"
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename="attach.zip"

UEsDBBQAAAAIABp6pzAZ29QhIQAAADoAAAAKAAAAYXR0YWNoLnR4dAvJyCx
ihKFtMycVD1erhA8crxcAFBLAQIUABQAAAAIABp6pzAZ29QhIQAAADoAAAA
gQAAAABhdHRhY2gudHh0UEsFBgAAAAABAAEAOAAAAEkAAAAAAA==

----=_NextPart_000_abcdef--
.
```

ColdFusion SMTP Tags

- <CFMAIL>
  Sends an email message that optionally contains query output, using an SMTP server.
- <CFMAILPARAM>
  Attaches a file or adds a header to an email message. Can only be used in the cfmail tag. You can use more than one cfmailparam tag within a cfmail tag.
- <CFMAILPART>
  Specifies one part of a multipart email message. Can only be used in the cfmail tag. You can use more than one cfmailpart tag within a cfmail tag. New in CFMX 6.1.

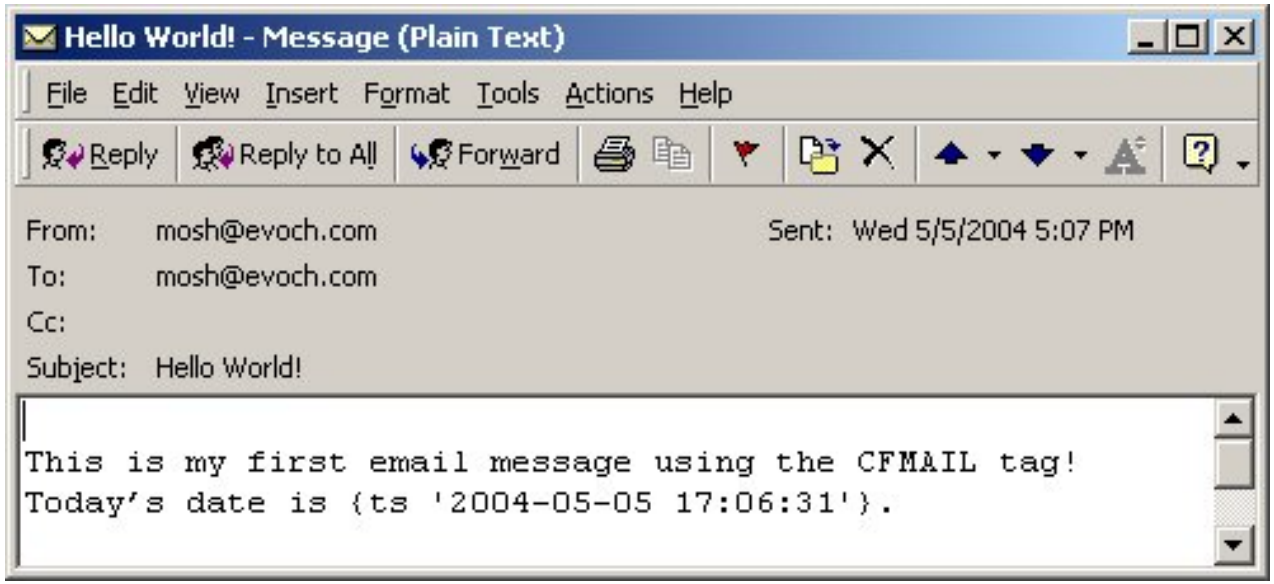
<CFMAIL>

Sends an email message that optionally contains query output, using an SMTP server. Most common attributes are below:

Attribute	Description
To, From, Subject	Required. To/From email addresses (with optional display names) and subject of the email message.
CC, BCC, replyTo	Optional. CC/BCC/reply to email addresses (with optional display names).
Username, Password	Optional. Used to send email via SMTP servers that require authentication.
Query, Group, startRow, maxRows	Optional. Used to send multiple messages with query-specific data in attributes or contents.
Server, Port	Optional. Specifies server and port to connect to. Overrides the value(s) set in the CF Administrator.

<CFMAIL> Example: Hello World

```
<CFMAIL TO="mosh@evoch.com" FROM="mosh@evoch.com"
  SUBJECT="Hello World!">
      This is my first email message using the CFMAIL tag!
      Today's date is #Now()#.
</CFMAIL>
```



<CFMAIL> Example: X-Mailer

```
<CFMAIL TO="mosh@evoch.com" FROM="mosh@evoch.com"
  SUBJECT="MAILERID attribute" MAILERID="Mosh Mail 2004">
      This email message was sent with the "Mosh Mail
      2004" email client.
</CFMAIL>
```

```
Message-ID: <16864859.1083955055111.JavaMail.SYSTEM@gambit>
Date: Fri, 7 May 2004 14:37:35 -0400 (EDT)
From: mosh@evoch.com
To: mosh@evoch.com
Subject: MAILERID attribute
X-Mailer: Mosh Mail 2004

This email message was sent with the "Mosh Mail 2004" email
client.
```

<CFMAILPARAM>

Attaches a file or adds a header to an email message. Can only be used in the cfmail tag. You can use more than one cfmailparam tag within a cfmail tag.

Attribute	Description
File	Attaches file to a message. Mutually exclusive with name attribute. The file is MIME encoded before sending.
Type	Optional. The MIME type of the attachment.
Name	Name of header. Case-insensitive. Mutually exclusive with file attribute.
Value	Optional. Value of the header.

<CFMAILPARAM> Example: X-Mailer #2

```
<CFMAIL TO="mosh@evoch.com" FROM="mosh@evoch.com"
  SUBJECT="CFMAILPARAM Tag">
      <CFMAILPARAM NAME="X-Mailer" VALUE="Mosh Mail 2004">
      This email message was sent with the "Mosh Mail
      2004" email client.
</CFMAIL>
```

```
Date: Fri, 7 May 2004 14:37:36 -0400 (EDT)
From: mosh@evoch.com
To: mosh@evoch.com
Subject: CFMAILPARAM Tag
X-Mailer: ColdFusion MX Application Server
X-Mailer: Mosh Mail 2004

This email message was sent with the "Mosh Mail 2004" email
client.
```

<CFMAILPARAM> Example: Importance

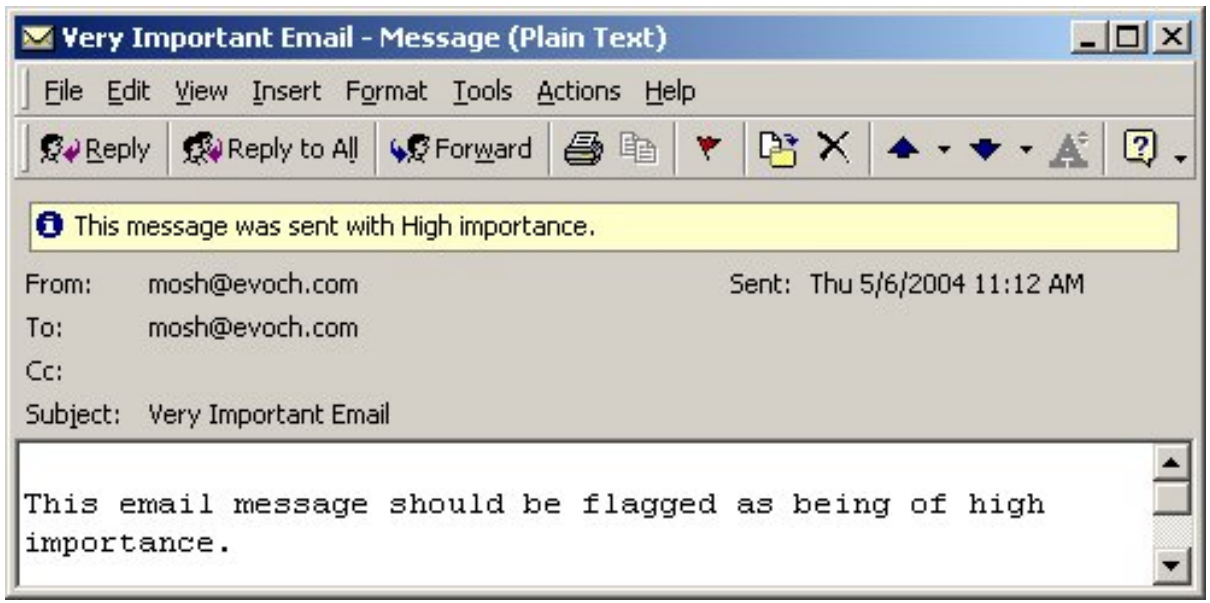
```
<CFMAIL TO="mosh@evoch.com" FROM="mosh@evoch.com"
  SUBJECT="Very Important Email">
      <CFMAILPARAM NAME="Importance" VALUE="High">
      This email message should be flagged as being of
      high importance.
</CFMAIL>
```

```
Date: Fri, 7 May 2004 14:37:18 -0400 (EDT)
From: mosh@evoch.com
To: mosh@evoch.com
Subject: Very Important Email
X-Mailer: ColdFusion MX Application Server
Importance: High

This email message should be flagged as being of high
importance.
```

<CFMAILPARAM> Example: Importance

```
<CFMAIL TO="mosh@evoch.com" FROM="mosh@evoch.com"
  SUBJECT="Very Important Email">
      <CFMAILPARAM NAME="Importance" VALUE="High">
      This email message should be flagged as being of
      high importance.
</CFMAIL>
```

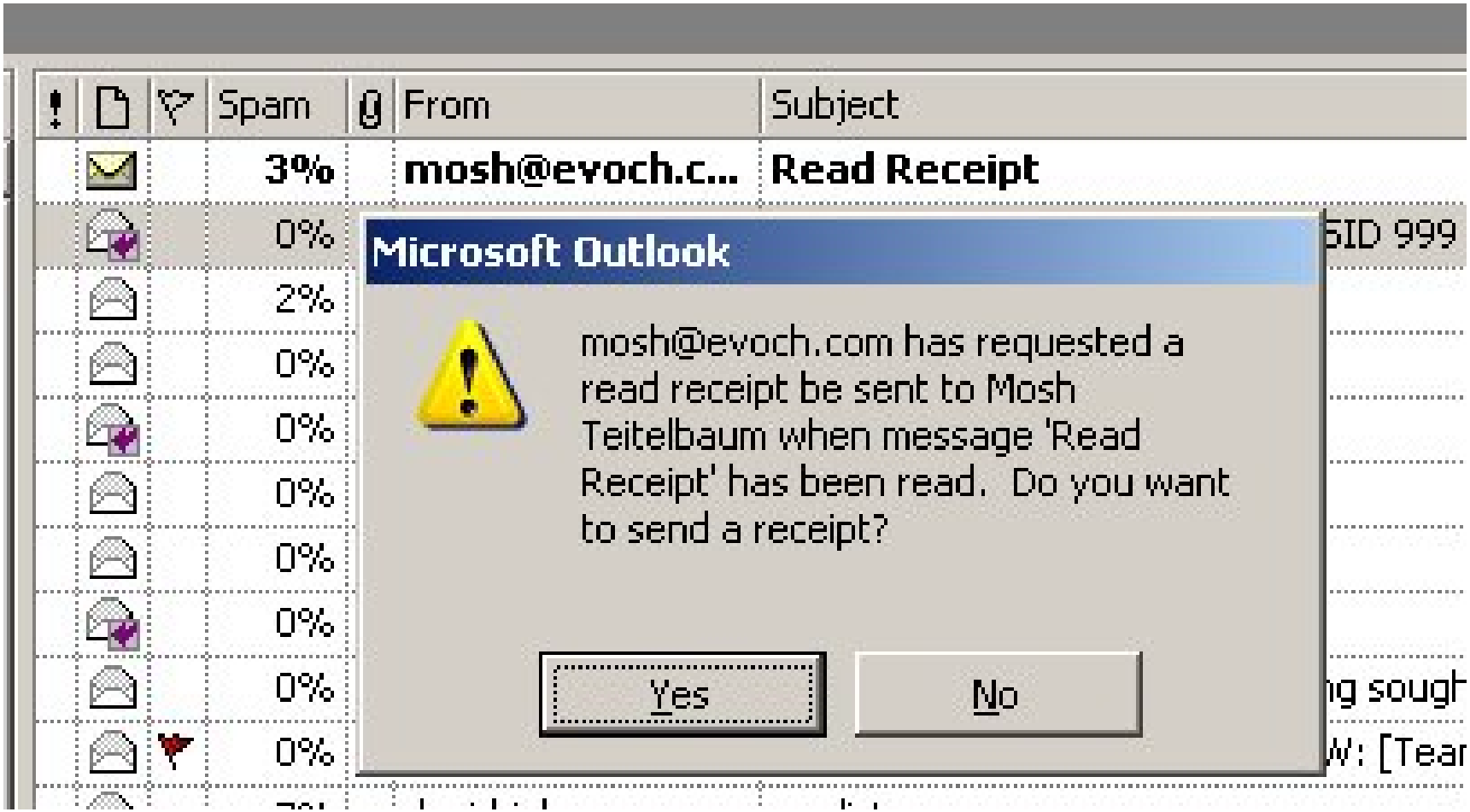


<CFMAILPARAM> Example: Read Receipt

```
<CFMAIL TO="mosh@evoch.com" FROM="mosh@evoch.com"
  SUBJECT="Read Receipt Email">
      <CFMAILPARAM NAME="Disposition-Notification-To"
      VALUE='"Mosh Teitelbaum" <mosh.teitelbaum@evoch.com>'>
      This email message should prompt you to send a
      return receipt.
</CFMAIL>
```

```
Date: Fri, 7 May 2004 15:50:41 -0400 (EDT)
From: mosh@evoch.com
To: mosh@evoch.com
Subject: Read Receipt
Disposition-Notification-To: "Mosh Teitelbaum"
➡   <mosh@evoch.com>
```

<CFMAILPARAM> Example: Read Receipt



<CFMAILPARAM> Example: X-Message

```
<CFMAIL TO="mosh@evoch.com" FROM="mosh@evoch.com"
  SUBJECT="Outlook Message">
      <CFMAILPARAM NAME="X-Message-Flag"
      VALUE="Visit www.evoch.com for more info!">
      Do you see my message?
</CFMAIL>
```

```
Date: Fri, 7 May 2004 16:54:41 -0400 (EDT)
From: mosh@evoch.com
To: mosh@evoch.com
Subject: Outlook Message
X-Message-Flag: Visit www.evoch.com for more info!
```

<CFMAILPARAM> Example: X-Message

<CFMAILPARAM> Example: Attachments

```
<CFMAIL TO="mosh@evoch.com" FROM="mosh@evoch.com"
  SUBJECT="Attachments">
      <CFMAILPARAM FILE="attach.txt" TYPE="plain/text">
      <CFMAILPARAM FILE="attach.zip" TYPE="application/zip">
      This email message has 2 attachments
</CFMAIL>
```

```
----=_NextPart_000_abcdef
Content-Type: application/zip; name="attach.zip"
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename="attach.zip"

UEsDBBQAAAAIABp6pzAZ29QhIQAAADoAAAAKAAAAYXR0YWNoLnR4dAvJyCx
ihKFtMycVD1erhA8crxcAFBLAQIUABQAAAAIABp6pzAZ29QhIQAAADoAA==

----=_NextPart_000_abcdef--
```

<CFMAILPART>

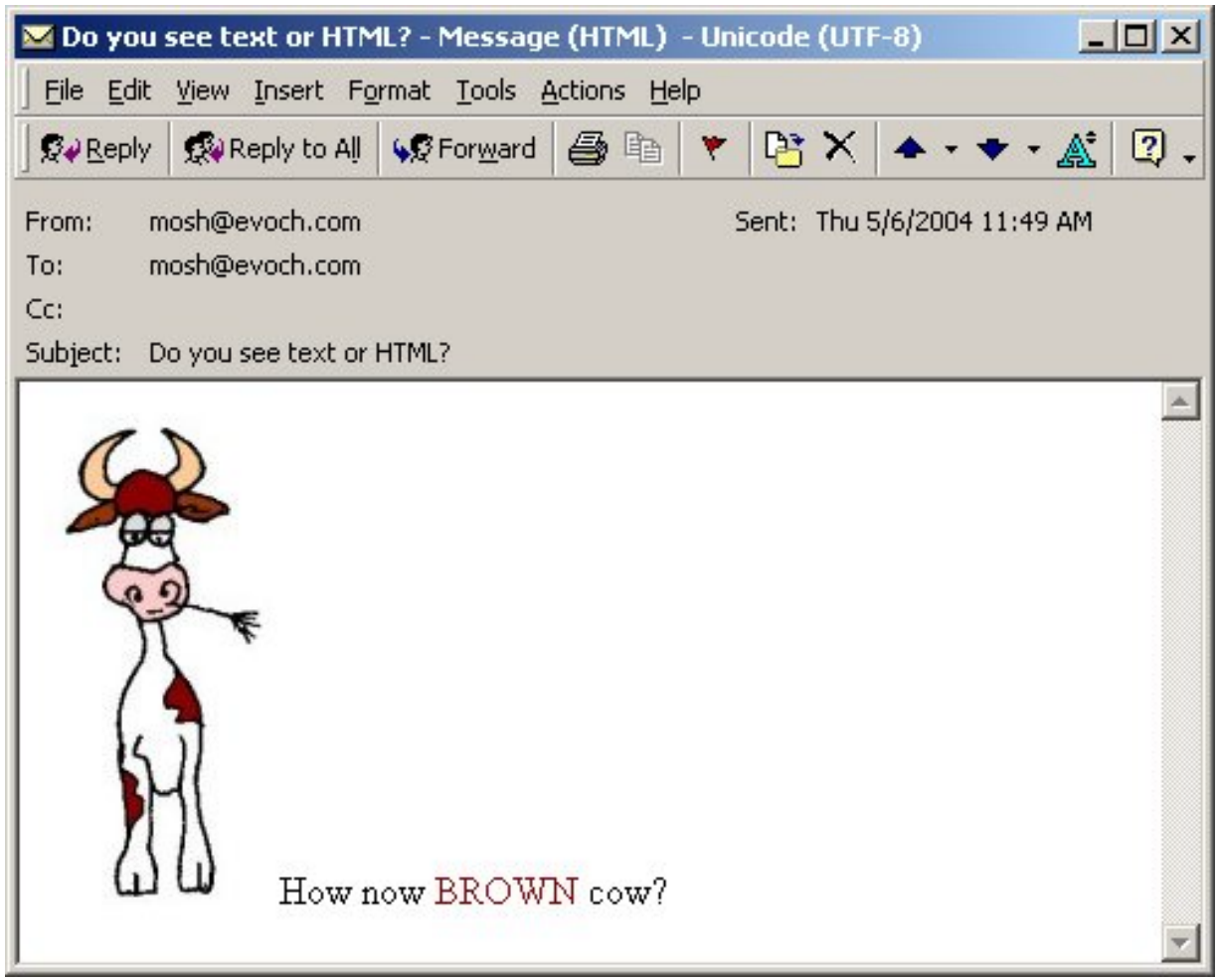
Specifies one part of a multipart email message. Can only be used in the cfmail tag. You can use more than one cfmailpart tag within a cfmail tag. New in CFMX 6.1

Attribute	Description
Type	Required. The MIME media type of the part. Can be a can be valid MIME media type or “text,” “plain,” or “html.”
WrapText	Optional. Specifies the maximum line length, in characters, of the mail text. If a line has more than the specified number of characters, replaces the last white space character, such as a tab or space, preceding the specified position with a line break. If there are no white space characters, inserts a line break at the specified position. A common value for this attribute is 72.
Charset	Optional. The character encoding in which the part text is encoded.

<CFMAILPART> Example: Basic HTML

```
<CFMAIL TO="mosh@evoch.com" FROM="mosh@evoch.com"
  SUBJECT="Do you see text or HTML?">
      <CFMAILPART TYPE="text/plain">
            How now brown cow?
      </CFMAILPART>
      <CFMAILPART TYPE="text/html">
            <IMG SRC="http://www.evoch.com/browncow.jpg">
            How now <FONT COLOR="##900000">BROWN</FONT> cow?
      </CFMAILPART>
</CFMAIL>
```

<CFMAILPART> Example: Basic HTML



SMTP Resources

- RFCs from http://www.ietf.org/rfc/rfc####.txt:
  - rfc821.txt – “Simple Mail Transfer Protocol”
  - rfc822.txt – “Standard for the format of ARPA Internet text messages”
  - rfc2821.txt – “Simple Mail Transfer Protocol”
  - rfc2822.txt – “Internet Message Format”
  - rfc1891.txt – “SMTP Service Extension for Delivery Status Notifications”
  - rfc1521.txt – “MIME (Multipurpose Internet Mail Extensions) Part One: Mechanisms for Specifying and Describing the Format of Internet Message Bodies”
  - rfc2045.txt - “Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies”
  - rfc2554.txt - “SMTP Service Extension for Authentication”

- SpamSource addin for MS Outlook – http://www.daesoft.com/SpamSource/

Closing

- Questions?

- Contact Info

Mosh Teitelbaum
evoch, LLC
mosh.teitelbaum@evoch.com
http://www.evoch.com/

- Extras
  - SMTP Status Code Meanings: *x*
  - SMTP Status Code Meanings: *y*
  - Defined SMTP Status Codes
  - Defined SMTP Headers

Extras: SMTP Status Code Meanings: *x*

The first digit denotes whether the response is good, bad, or incomplete

Range	Meaning
1yz	Positive Preliminary – The command has been accepted pending confirmation of the information included in the reply.
2yz	Positive Completion – The command has been accepted.
3yz	Positive Intermediate – The command has been accepted pending receipt of additional information (such as with the DATA command).
4yz	Transient Negative Completion – The command was not accepted but the reason for non-acceptance is temporary and the client may try again.
5yz	Permanent Negative Completion – The command was not accepted and will never be accepted as currently structured.

Extras: SMTP Status Code Meanings: *y*

Specifies the type of error (syntax, information, connections, mail system)

Range	Meaning
x0z	Syntax – Syntax errors, syntactically correct commands that do not fit any functional category, and unimplemented commands
x1z	Information – Replies to requests for information such as status or help
x2z	Connections – Replies referring to the connection
x3z, x4z	Unspecified
x5z	Mail System – Indicates the status of the server

Extras: Defined SMTP Status Codes

Code	Meaning	Code	Meaning
211	System status or system help reply	452	Requested action not taken: insufficient system storage
214	Help message	500	Syntax error, command unrecognized
220	<domain> Service ready	501	Syntax error in parameters or arguments
221	<domain> Service closing transmission channel	502	Command not implemented
250	Requested mail action OK, completed	503	Bad sequence of commands
251	User not local; will forward to <forward-path>	504	Command parameter not implemented
252	Cannot VRFY user, but will accept message and attempt delivery	550	Requested action not taken: mailbox unavailable
354	Start mail input; end with <CRLF>.<CRLF>	551	User not local; please try <forward-path>
421	<domain> Service not available, closing transmission channel	552	Requested mail action aborted: exceeded storage allocation
450	Requested mail action not taken: mailbox unavailable	553	Requested action not taken: mailbox name not allowed
451	Requested action aborted: local error in processing	554	Transaction failed

Extras: Defined SMTP Headers

Header	Description	Header	Description
Return-Path, Received	Used to trace a messages progress from sender to receiver(s)	Reply-To	Address to which replies should be sent
Resent-Date	Resent fields should be added to any message that is reintroduced by a user into the transport system. A separate set of resent fields should be added each time this is done. All of the resent fields corresponding to a particular resending of the message should be together. Each new set of resent fields is prepended to the message; that is, the most recent set of resent fields appear earlier in the message. No other fields in the message are changed when resent fields are added.	To	Address(es) of primary recipient(s)
Resent-From		Cc	Address(es) of auxiliary recipient(s)
Resent-Sender		Bcc	Address(es) of “hidden” recipient(s)
Resent-To		Message-ID	A single, unique identifier
Resent-Cc		In-Reply-To	Specifies the Message-IDs of the message(s) this message is in reply to
Resent-Bcc		References	Specifies the Message-IDs of the message(s) this message refers to
Resent-Message-ID		Subject	A short string identifying the topic of the message
Date	Date/time the mail was originally sent	Comments	Comments on the body of the message
From	Specifies the author(s) of the mail	Keywords	A comma-separated list of important words and phrases
Sender	Specifies the sender’s address. Req’d if From has more than 1 address.	<custom>	Custom headers should begin with “X-” (X dash)