

# Should I Write HTML or Plain Text Emails?

Posted At : July 4, 2008 12:36AM | Posted By : Matt Quackenbush

Related Categories: ColdFusion

One of the questions I have always asked myself when writing an application that sends out emails (most applications do) is: "should I write HTML or plain text emails?" As you know, most but not all email clients support HTML emails, and still other users disable HTML emails for one reason or another.

"Yeah," you say, "that's why ColdFusion has the 'type' attribute on the <cfmail> tag." And you are absolutely correct in saying so. But that doesn't solve the problem of how to *write* the email in your code; it only solves the problem of how to *send* the email.

## The Problem: 2 Emails or Garbled Display

You see, if you write the email using (x)HTML and then send it in plain text, the recipient ends up with a horribly garbled email message filled with markup meant for display that they have to try and read around. Many people will simply give a quick click of the delete button rather than try and decipher what they're supposed to read vs. what's supposed to make it look pretty (but fails miserably in doing so in instances like this).

Years ago I actually went so far as to write an HTML *and* a plain text version, and then use a <cfif> to determine which one to send to that particular recipient. After a short time of doing that, I said "screw that" and just wrote an HTML version. Luckily I have been in a position where 99+% of my users actually view their email as HTML, so I could get away with it. Well, I am now working on a project where it is anticipated that there will be a fairly equal split between HTML and plain text recipients. So, once again, I was faced with that age-old question: HTML or plain text?

## The Goal

Simply stated, the goal is to write one single email message, and have it nicely formatted and readable by both HTML-enabled email clients and plain text only email clients.

## The Solution: setBodyAsText()

In my databases, I have a bit (boolean) column on the email address table named 'acceptHTML'. If set to true, then all emails sent by the application to that address will be sent as HTML. If set to false, then they get sent as plain text. As mentioned previously, this is handled nicely by the 'type' attribute on the <cfmail> tag. The other half of the equation is the setBodyAsText() method in my EmailService:

```

.. <cffunction name= "setBodyAsText"
?. hint= "I format the email body as plain text for sending to email addresses that do not support HTML"
?. returnType= "void"
?. output= "no"
?. access= "private" >
?. <cfargument name= "email"
?. hint= "The email object"
?. required= "yes"
?. type= "any" />
.. <cfscript>
?. var body = email.getBody();
?. body = reReplace (body, " </p> [^<]* <p> " , chr( 13 ), "all" );
?. body = replace (body, " <hr /> " , repeatString( "=", 70 ), "all" );

```

```

5. body = reReplace (body, "#chr( 9 )#|<(/*)[^>]*>" , "" , "all" );
6. email.setBody(body);
7. </cfscript>
8. </cffunction>

```

What does this code do? Let's break it down real quick:

Line 6: Pass in our Email object

Line 12: Set a local variable to the value of getBody()

Line 13: Replace a closing paragraph tag followed by an opening paragraph tag with a carriage return

Line 14: Replace a horizontal rule with a string of equal signs, forming a plain text horizontal rule

Line 15: Replace tabs and any other HTML tags with an empty string

Line 16: Set the new, plain text body back to the Email object

That's it. Problem solved. I now have one single email message written - in (x)HTML - and can quickly and efficiently send it either as HTML or plain text.

"Hey, Quack, I don't even know what an 'EmailService' is. Can't I just use setBodyAsText() as a UDF in my application?"

Sure, a slight modification and you're golden.

```

.. <cffunction name= "setBodyAsText"
?.. hint= "I format the email body as plain text for sending to email addresses that do not support HTML"
?.. returntype= "string"
?.. output= "no"
?.. access= "private" >
?.. <cfargument name= "body"
?.. hint= "The email body"
?.. required= "yes"
?.. type= "string" />
.. <cfscript>
?.. body = reReplace (body, " </p> [^<]* <p> " , chr( 13 ), "all" );
?.. body = replace (body, " <hr /> " , repeatString( "=", 70 ), "all" );
?.. body = reReplace (body, "#chr( 9 )#|<(/*)[^>]*>" , "" , "all" );
?.. return body;
?.. </cfscript>
?.. </cffunction>

```

Now all you do is pass the body of your email into the method, and it will return it to you formatted in plain text.

"That looks simple enough, but can I see it in action?"

Ayep. [View the demo](#) .