Email

Component Overview

{{ http://commons.apache.org/email/images/email-logo-white.png }}	Commons Email aims to provide a API for sending email. It is built on top of the JavaMail API, which it aims to simplify. A lot of information is available on the Commons Email website. If you don't find the information you need you can always contact us using one of the mailing lists.

(Official) Project Status

The current status (updated 10/2007) is available through SVN here. The 1.1 release has been approved and announced. The Commons community welcomes those who wish to drive the project further.

Release Plan

No current plan is active as the 1.1 release has been approved and announced.

Archived discussion of 1.1 changes

This section contains notes on the reasoning behind the 1.1 fixes and waived changes so that other developers can use it as a starting point for their work.

New features

(Added in rev 545815) EMAIL-35: Allow DataSources to be directly embedded in HtmlEmails.

Bug fixes

email 1.0 doesn't handle HTML embedding or attachments correctly. This is really bad, and is reason enough for a 1.1 release.

(Added in rev 510708) EMAIL-50: HTML mail doesn't display correctly (the issue says in Outlook 2000, but the display fails on every common mail client I tried). It's not an Outlook-specific error. A patch is available that fixes this and has been tested on several different clients; also, the new patch adheres properly to the MIME specs in RFC 2045-2049 where email 1.0 did not. The fix also resolves two other open issues:

- EMAIL-28: HTML attachments don't work correctly.
- EMAIL-52: Don't embed the same URL multiple times in the same email.

Character set fixes/support

email 1.0 provided limited support for different charsets, and it's generated four issues.

(Added in rev 510275) EMAIL-1: Default email charset not used for text content. A patch has been uploaded that defines three different cases for this behavior and tests/fixes each.

(Added in rev 510715) EMAIL-54: Propose a new Charset enum. The current patch uses the JDK 1.4 java.nio.charset.Charset class to validate user-supplied charset names. This removes the need for a separate enumeration of "supported" classes and greatly reduces the headache of maintaining the support as the JVM continues to evolve. This also incorporates fixes for two other open issues:

- EMAIL-25: Can't set charset for a single address.
- EMAIL-14: Support a couple specific charsets.

Build fixes/enhancements

(Added in rev 510706) EMAIL-62: Enforce 1.4 source/bytecode in ant and maven 1 builds

(Added in rev 510704) EMAIL-63: Submitted maven 2 pom.xml.

(Added in rev 544629) EMAIL-64: use a different test email server from a live project, not a dead one. Patch available to use wiser from the subethasmtp project. The test cases have been ported and the wiser packages uploaded to maven2 for the enjoyment of all.

Waived features

(Waived for 1.1) EMAIL-56: Support creating Email subclasses from MimeMessages.

(BenSpeakmon) One of the MimeMessage constructors in JavaMail (both 1.3 and 1.4) already does this, BTW. I'm not sure this is something that falls within commons-email's scope. The commons-email API, I think, is for the common cases where you just want to build and send an email without needing the power (or complexity) of JavaMail. If you're already pulling messages from an email server, I don't know why you wouldn't just use JavaMail for manipulating it – the power and complexity is just what you need for those kinds of jobs. And it doesn't seem worth the trouble to duplicate any of that code in commons-email when the existing code works just fine. I'd recommend WONTFIXing this one.

EMAIL-6: Allow attaching one subclass of Email to another.

_(BenSpeakmon) I'm not convinced that it's useful to support attaching one subclass of Email to another subclass – that is, the original report creates an HtmlEmail and then wants to attach that to a MultiPartEmail. The current design of commons-email would make this very tricky, but aside from that I don't see the general usefulness of doing so. One case where I do see a use for this would be when the user wants to attach a MimeMessage he got from somewhere (like a server or a filestore) to a subclass of Email. This would mean adding MultiPartEmail.addMimeMessage() methods which would then be attached like the current addPart() methods. (We'd have to have different names, since MimeMessage doesn't share a common interface ancestor with MimeMultipart.) There's still a problem, though; such a MimeMessage would have arbitrarily complex content in it. It could be multipart, HTML text, full of attachments, etc., and there's no way to know what really needs to be attached.

I think that this has to fall into that class of problems that would be better served with JavaMail, so I'd recommend this be wontfixed._

EMAIL-65: Improve HtmlEmail MIME layout.

(BenSpeakmon) This could be made better as detailed in the issue – I took a crack at it, but it turned out to touch a lot of moving parts to the point where I felt uncomfortable doing it in a point release. If anybody wants to use this as a starting point for a 2.0 release, possibly based on the contributed code, feel free 🙂