1 简述.....1
2 配置.....1
2.1 全局配置.....1
2.1.1 全局属性详解.....3
2.1.2 全局邮件变量.....3
2.2 项目配置.....5
2.2.1 项目基本配置.....6
2.2.1.1 项目高级配置.....6
2.2.1.2 触发器类型.....7
2.2.1.3 项目邮件变量.....7
3 Jelly 脚本.....7
4 总结.....9
5 参考资料.....9

使用 email-ext 替换 Jenkins 的默认邮件通知

1 简述

众所周知，Jenkins 默认提供了一个邮件通知，能在构建失败、构建不稳定等状态后发送邮件。但是它本身有很多局限性，比如它的邮件通知无法提供详细的邮件内容、无法定义发送邮件的格式、无法定义灵活的邮件接收配置等等。在这样的情况下，我们找到了 Jenkins Email Extension Plugin。该插件能允许你自定义邮件通知的方方面面，比如在发送邮件时你可以自定义发送给谁，发送具体什么内容等等。本文不会告诉你如何安装该插件，关于插件的安装请参考这里。

2 配置

它主要包含两个部分：全局配置和项目配置。

2.1全局配置

当然，在一个项目中应用 email-ext 插件之前，您必须做一些全局的配置。现在先跳转到 Jenkins 的“系统设置”页面，如下图：

找到标题为“**Extended E-mail Notification**”的片段，你就能配置一些全局的 email-ext 属性。这些属性必须匹配你 SMTP 邮件服务器的设置。这一节不仅能配置成 Jenkins 原有邮件通知的镜像(虽然有很多配置是一样的，但这是个不同的扩展点)，而且还增加了一些额外的功能。输入框中名为 Default Subject 和 Default Content 的项允许你在全局级别配置邮件的内容。这样做的话，可以使您为所有的项目按您的需求做更好的、更简单的配置。如下图。

释放个我的配置：

Default Subject：构建通知:$PROJECT_NAME - Build # $BUILD_NUMBER - $BUILD_STATUS!

Default Content：

<hr/>

(本邮件是程序自动下发的，请勿回复！)
<hr/>

项目名称：$PROJECT_NAME
<hr/>

构建编号：$BUILD_NUMBER
<hr/>

svn 版本号：${SVN_REVISION}
<hr/>

构建状态：$BUILD_STATUS
<hr/>

触发原因：${CAUSE}
<hr/>

构建日志地址：${BUILD_URL}console
<hr/>

构建地址：$BUILD_URL
<hr/>

变更集:${JELLY_SCRIPT,template="html"}
<hr/>

下面解释一下常用的属性。

2.1.1 全局属性详解

1. Override Global Settings：如果不选，该插件将使用默认的 E-mail Notification 通知选项。反之，您可以通过指定不同于(默认选项)的设置来进行覆盖。
2. Default Content Type：指定构建后发送邮件内容的类型，有 Text 和 HTML 两种.
3. Use List-ID Email Header：为所有的邮件设置一个 List-ID 的邮件信头，这样你就可以在邮件客户端使用过滤。它也能阻止邮件发件人大部分的自动回复(诸如离开办公室、休假等等)。你可以使用你习惯的任何名称或者 ID 号，但是他们必须符合如下其中一种格式(真实的 ID 必须要包含在<和>标记里)：

   <ci-notifications.company.org>

   Build Notifications <ci-notifications.company.org>

   "Build Notifications" <ci-notifications.company.org>

   关于更详细的 List-ID 说明请参阅 RFC-2919.
4. Add 'Precedence: bulk' Email Header：设置优先级,更详细说明请参阅 RFC-3834.
5. Default Recipients：自定义默认电子邮件收件人列表。如果没有被项目配置覆盖,该插件会使用这个列表。您可以在项目配置使用$ DEFAULT_RECIPIENTS 参数包括此默认列表，以及添加新的地址在项目级别。添加抄送：cc:电子邮件地址例如,CC:someone@somewhere.com
6. Reply To List：回复列表, A comma separated list of e-mail addresses to use in the Reply-To header of the email. This value will be available as $DEFAULT_REPLYTO in the project configuration.
7. Emergency reroute：如果这个字段不为空，所有的电子邮件将被单独发送到该地址（或地址列表）。
8. Excluded Committers：防止邮件被邮件系统认为是垃圾邮件,邮件列表应该没有扩展的账户名(如:@domain.com),并且使用逗号分隔
9. Default Subject：自定义邮件通知的默认主题名称。该选项能在邮件的主题字段中替换一些参数，这样你就可以在构建中包含指定的输出信息。
10. Maximum Attachment Size：邮件最大附件大小。
11. Default Content：自定义邮件通知的默认内容主体。该选项能在邮件的内容中替换一些参数，这样你就可以在构建中包含指定的输出信息。
12. Default Pre-send Script：默认发送前执行的脚本（注：grooy 脚本，这是我在某篇文章上看到的，不一定准确）。
13. Enable Debug Mode：启用插件的调试模式。这将增加额外的日志输出，构建日志以及 Jenkins 的日志。在调试时是有用的，但不能用于生产。
14. Enable Security：启用时，会禁用发送脚本的能力，直接进入 Jenkins 实例。如果用户试图访问 Jenkins 管理对象实例，将抛出一个安全异常。
15. Content Token Reference：邮件中可以使用的变量，所有的变量都是可选的。具体介绍请查看全局邮件变量章节。

2.1.2 全局邮件变量

Ps：看着感觉有点晕头，对比着 Jenkins 页面看要好些。

email-ext 插件允许使用**变量**来动态插入数据到邮件的主题和内容主体中。**变量**是一个以$(美元符号)开始，并以空格结束的字符串。当一个邮件触发时，主题和内容主体字段的所有变量都会通过真实的值动态地替换。同样，变量中的"值"能包含其它的变量，都将被替换成真实的内容。

比如，项目配置页的默认主题和内容分别对应的是全局配置页面的 DEFAULT_SUBJECT 和 DEFAULT_CONTENT，因此它会自动地使用全局的配置。同理，触发器中的 Subject 和 Content 分别对应的是项目配置页面的 DEFAULT_SUBJECT 和 DEFAULT_CONTENT，所以它也会自动地使用项目的配置。由于变量中的"值"能包含其它的变量，所以就能为变量快速地创建不同的切入点：全局级别(所有项目)，专属级别(单一项目)，触发器级别(构建结果)。

如果你要查看所有可用的变量，你可以点击配置页的 Content Token Reference 的**问号**获取详细的信息。

所有的变量都是可选的，每个变量可以如下表示，字符串类型使用 name="value"，而布尔型和数字型使用 name=value。如果{和}标记里面没有变量，则不会被解析。示例：$TOKEN,${TOKEN},${TOKEN,count=100},${ENV,var="PATH"}
提示：用英文逗号分隔变量的参数。

下面我解释一下常用的属性。

- ${FILE,path="PATH"} 包括指定文件（路径）的含量相对于工作空间根目录。
    - path 文件路径，注意：是工作区目录的相对路径。
- ${BUILD_NUMBER} 显示当前构建的编号。
- ${JOB_DESCRIPTION} 显示项目描述。
- ${SVN_REVISION} 显示 svn 版本号。还支持 Subversion 插件出口的 SVN_REVISION_n 版本。
- ${CAUSE} 显示谁、通过什么渠道触发这次构建。
- ${CHANGES } -显示上一次构建之后的变化。
    - showPaths 如果为 true,显示提交修改后的地址。默认 false。
    - showDependencies 如果为 true，显示项目构建依赖。默认为 false
    - format 遍历提交信息，一个包含%X 的字符串，其中%a 表示作者，%d 表示日期，%m 表示消息，%p 表示路径，%r 表示版本。注意，并不是所有的版本系统都支持%d 和%r。如果指定 showPaths 将被忽略。默认"[%a] %m\\n"。
    - pathFormat 一个包含"%p"的字符串，用来标示怎么打印路径。
- ${BUILD_ID}显示当前构建生成的 ID。
- ${PROJECT_NAME} 显示项目的全名。（见 AbstractProject.getFullDisplayName）
- ${PROJECT_DISPLAY_NAME} 显示项目的显示名称。（见 AbstractProject.getDisplayName）
- ${SCRIPT} 从一个脚本生成自定义消息内容。自定义脚本应该放在"$JENKINS_HOME/email-templates"。当使用自定义脚本时会默认搜索$JENKINS_HOME/email-templatesdirectory 目录。其他的目录将不会被搜索。
    - script 当其使用的时候，仅仅只有最后一个值会被脚本使用（不能同时使用 script 和 template）。
    - template 常规的 simpletemplateengine 格式模板。
- ${JENKINS_URL} 显示 Jenkins 服务器的 url 地址（你可以再系统配置页更改）。
- ${BUILD_LOG_MULTILINE_REGEX}按正则表达式匹配并显示构建日志。
    - regex
      java.util.regex.Pattern 生成正则表达式匹配的构建日志。无默认值，可为空。
    - maxMatches 匹配的最大数量。如果为 0，将匹配所有。默认为 0。
    - showTruncatedLines 如果为 true，包含[...truncated ### lines...]行。默认为 true。
    - substText 如果非空，就把这部分文字（而不是整行）插入该邮件。默认为空。
    - escapeHtml 如果为 true，格式化 HTML。默认为 false。
    - matchedSegmentHtmlStyle 如果非空，输出 HTML。匹配的行数将变为 html escaped matched line 格式。默认为空。
- ${BUILD_LOG} 显示最终构建日志。
    - *maxLines* 日志最多显示的行数，默认 250 行。

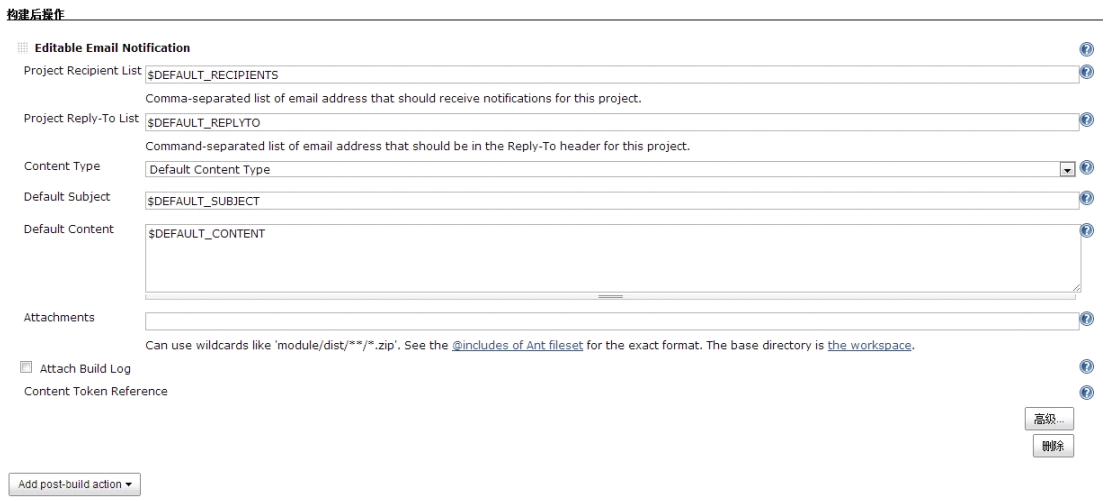
- *escapeHtml* 如果为 true，格式化 HTML。默认 false。

- ${PROJECT_URL} 显示项目的 URL 地址。

- ${BUILD_STATUS} -显示当前构建的状态(失败、成功等等)
- ${BUILD_URL} -显示当前构建的 URL 地址。
- ${CHANGES_SINCE_LAST_SUCCESS} -显示上一次成功构建之后的变化。
  - reverse 在顶部标示新近的构建。默认 false。
  - format 遍历构建信息，一个包含%X 的字符串，其中%c 为所有的改变，%n 为构建编号。默认”Changes for Build #%n\n%c\n”。
  - showPaths,changesFormat,pathFormat 分别定义如${CHANGES}的 showPaths、format 和 pathFormat 参数。
- ${CHANGES_SINCE_LAST_UNSTABLE} -显示显示上一次不稳固或者成功的构建之后的变化。
  - reverse 在顶部标示新近的构建。默认 false。
  - format 遍历构建信息，一个包含%X 的字符串，其中%c 为所有的改变，%n 为构建编号。默认”Changes for Build #%n\n%c\n”。
  - showPaths,changesFormat,pathFormat 分别定义如${CHANGES}的 showPaths、format 和 pathFormat 参数。
- ${ENV} –显示一个环境变量。
  - var– 显示该环境变量的名称。如果为空，显示所有，默认为空。
- ${FAILED_TESTS} -如果有失败的测试，显示这些失败的单元测试信息。
- ${JENKINS_URL} -显示 Jenkins 服务器的地址。(你能在“系统配置”页改变它)。
- ${HUDSON_URL} -不推荐，请使用$JENKINS_URL
- ${PROJECT_URL} -显示项目的 URL。
- ${SVN_REVISION} -显示 SVN 的版本号。
- ${JELLY_SCRIPT} -从一个 Jelly 脚本模板中自定义消息内容。有两种模板可供配置：HTML 和 TEXT。你可以在 $JENKINS_HOME/email-templates 下自定义替换它。当使用自动义模板时，”template”参数的名称不包含 “.jelly”。
  - template 模板名称，默认”html”。
- ${TEST_COUNTS} -显示测试的数量。
  - var– 默认“total”。
    - total -所有测试的数量。
    - fail -失败测试的数量。
    - skip -跳过测试的数量。

2.2项目配置

要想在一个项目中使用 email-ext 插件，你首先必须在项目配置页激活它。在**构建后操作——**”Add Post-build Actions”选项中勾选”Editable Email Notification”标签。



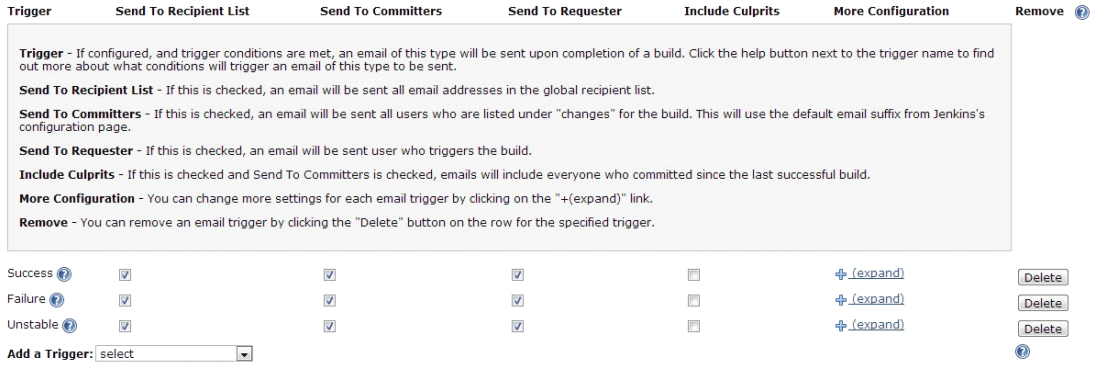
2.2.1 项目基本配置

当插件激活后你就能编辑如下字段（只列出常用的字段）：

- **Project Recipient List**：这是一个以逗号(或者空格)分隔的收件人邮件的邮箱地址列表。允许您为每封邮件指定单独的列表。Ps：如果你想在默认收件人的基础上添加收件人：$DEFAULT_RECIPIENTS,<新的收件人>
- **Default Subject**：允许你配置此项目邮件的主题。
- **Default Content**：跟 Default Subject 的作用一样，但是是替换邮件内容。
- **Attach Build Log**：附件构建日志。
    - **Compress Build Log before sending**：发送前压缩生成日志（zip 格式）。

2.2.1.1项目高级配置

要查看插件的高级配置，请点击"**高级**"按钮。该选项允许您各种类型的邮件触发器指定接收者。默认情况下，是没有配置的触发器，所以默认情况下不会发送邮件。要增加更多的触发器，选择"**Add a Trigger**"旁边下拉列表中的类型，它会增加到控件上面的列表中。一旦你增加了一个触发器，你就可以对它做一些选择。如果你点击一个触发器旁边的"**?**"号，它将告诉你在什么条件下会触发邮件发送。如下图。



- **Send to Recipient List**：如果勾选，邮件将发送到"**Project Recipient List**"中的所有邮件地址。

- **Send to Committers**：该邮件会发给上次构建时检查过代码的人员，该插件会基于提交者的 ID 和追加 Jenkins 配置页面的(default email suffix)默认邮件后缀来生成一个邮件地址。譬如，上次提交代码的人是"first.last"， 默认的电子邮件后缀为"@somewhere.com"，那么电子邮件将被发送到"first.last@ somewhere.com"。
- **Send To Requester**：如果勾选，邮件将发送给构建触发者。
- **Include Culprits**：如果勾选，而且 "Send To Committers"勾选，邮件将包含最后成功构建的提交者。
- **More Configuration**：通过单击"**+(expand)**"链接您能为每个邮件触发器作更多单独的设置。
    - **Recipient List**：这是一个以逗号(或者空格)分隔的可接受邮件的邮箱地址列表。如果触发就发送邮件到该列表。该列表会追加在"Global Recipient List"里。
    - **Subject**：指定选择邮件的主题。注意：高级选项中的邮件触发器类型可覆盖对它的配置。
    - **Content**：指定选择邮件的内容主体。注意：高级选项中的邮件触发器类型可覆盖对它的配置。
- **Remove** 通过单击指定触发器当前行的"**Delete**"按钮，你可以删除该触发器。

2.2.1.2触发器类型

注意：所有的触发器都只能配置一次。

Failure：即时发送构建失败的邮件。如果"Still Failing"触发器已配置，而上一次构建的状态是"Failure"，那么"Still Failing"触发器将发送一封邮件来替代(它)。
Unstable：即时发送构建不稳固的邮件。如果"Still Unstable"触发器已配置，而上一次构建的状态是"Unstable"，那么"Still Unstable"触发器将发送一封邮件来替代(它)。
Still Failing：如果两次或两次以上连续构建的状态为"Failure"，发送该邮件。
Success：如果构建的状态为"Successful"发送邮件。如果"Fixed"已配置，而上次构建的状态为"Failure"或"Unstable"，那么"Fixed"触发器将发送一封邮件来替代(它)。
Fixed：当构建状态从"Failure"或"Unstable"变为"Successful"时发送邮件。
Still Unstable：如果两次或两次以上连续构建的状态为" Unstable "，发送该邮件。
Before Build：当构建开始时发送邮件。

2.2.1.3项目邮件变量

注意：这里只解释全局配置页面中缺少的变量。

- **${DEFAULT_SUBJECT} :** 这是 Jenkins 系统配置页面默认配置的邮件主题
- **${DEFAULT_CONTENT} :** 这是 Jenkins 系统配置页面默认配置的邮件内容主体
- **${PROJECT_DEFAULT_SUBJECT} :**这是项目的默认邮件主题。高级配置中使用该令牌的结果要优先于 Default Subject 字段。警告：不要在 Default Subject 或者 Default Content 中使用该令牌，它会产生一个未知的结果。
- **${PROJECT_DEFAULT_CONTENT} :** 这是项目的默认邮件内容主体。高级配置中使用该令牌的结果要优先于 Default Content 字段。警告：不要在 Default Subject 或者 Default Content 中使用该令牌，它会产生一个未知的结果。

3 Jelly 脚本

从 Jenkins 2.9 版本开始我们可以使用 Jelly 脚本。Jelly 脚本跟 Hudson 的 API 挂钩，能获得你想要的任何信息，所以它很强大。插件有两个打包后的 Jelly 脚本，当然你也可以自定义(脚本)。

关于插件中默认的两个 Jelly 脚本：一个用来设计 HTML 格式邮件，另一个则是定义 TEXT 格式邮件。你能通过使用模板参数指定插件调用哪一个脚本。它们的使用方法如下：

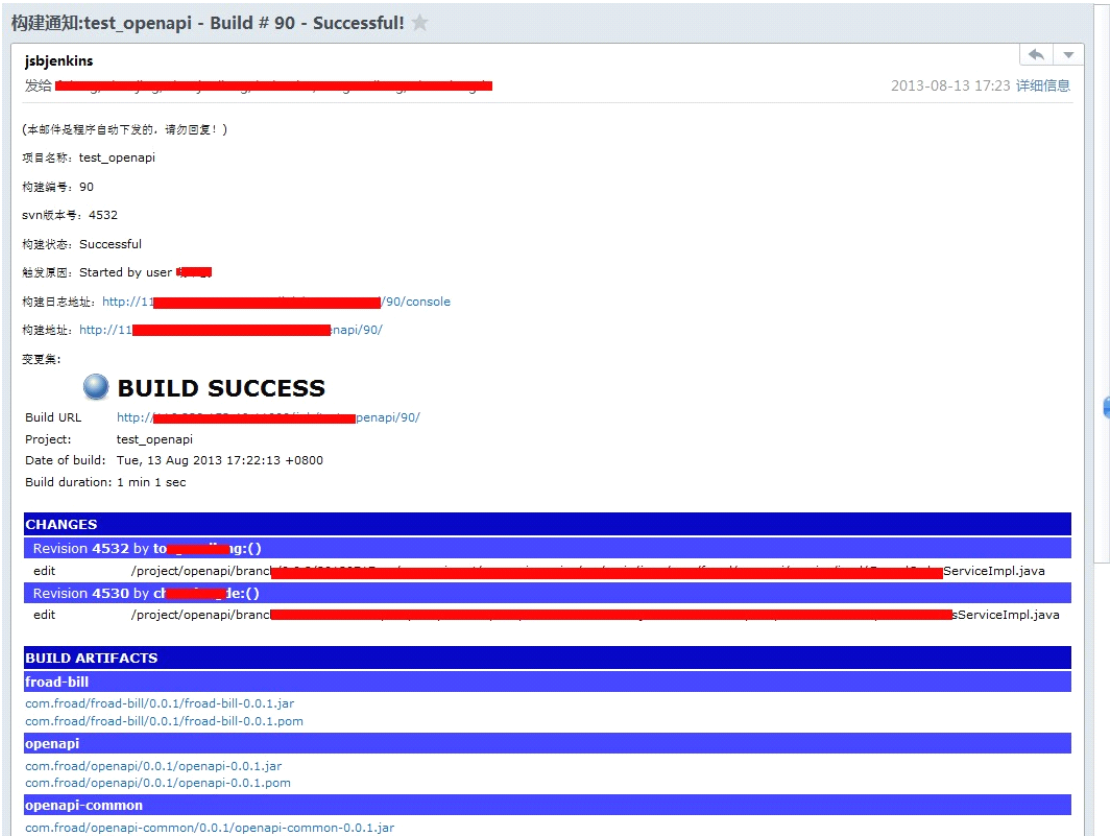
- 文本格式：${JELLY_SCRIPT,template="text"}
- HTML 格式:${JELLY_SCRIPT,template="html"}

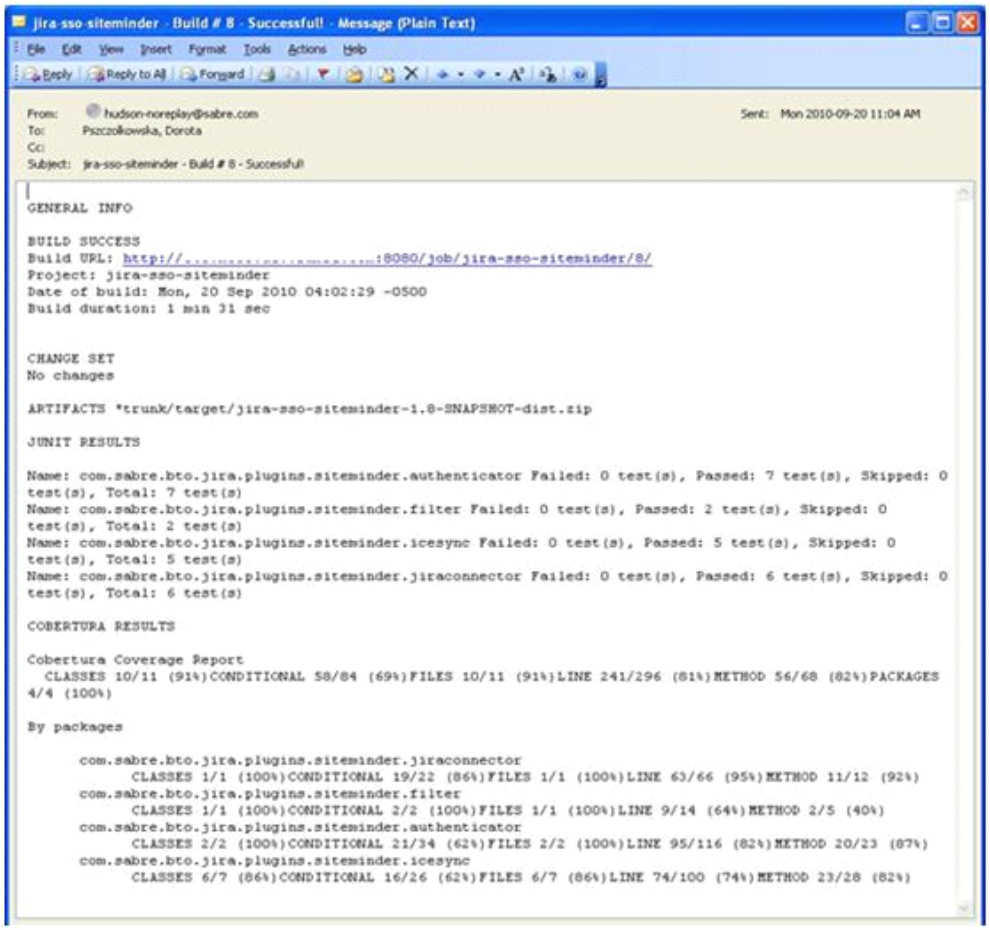
你也能编写属于自己的 Jelly 脚本。Jelly 脚本能跟 Jenkins 的 API(包括 hudson.model.AbstractBuild 和 hudson.model.AbstractProject)挂钩，因而特别强大。如果你打算这么做，你可以先参考现有的 html 和 text 脚本一探究竟。

值得注意的是，拥有 Hudson 管理员权限是使用自定义 Jelly 脚本(该脚本没有跟 email-ext 打包)的前提。脚本的生成步骤本身其实相对简单：

1. 创建 Jelly 脚本。脚本的名称应该是*<名称>.jelly*。名称以.jelly 结尾是很重要的。
2. 把脚本存放在 *JENKINS_HOME|email-templates* 文件夹里。
3. 使用 Jelly 变量，让 template 匹配你的脚本名称(不要包含后缀)。比如，脚本的名称为 foobar.jelly，则邮件内容中应该是${JELLY_SCRIPT,template="foobar"}。

下面两个图就是就是使用 Jelly 脚本生成的邮件（最新版 Email-ext 新增 html_gamil 模板，它跟 html 模板类似，所以这里不再显示它的截图）：





4 总结

以上就是我介绍的 Email-ext 插件，由于自己的局限，对于它的使用没有更深的了解。参考资料[2]中还有关于它的扩展，你也可以自行扩充它的功能。文章部分内容来源于参考资料[3]。如果您有关于该插件以及 Jenkins 使用的更多更好的感受，我期待与您一起分享。

5 参考资料

[1]《Maven 实战》第 11 章 11.9 邮件反馈。

[2]https://wiki.jenkins-ci.org/display/JENKINS/Email-ext+plugin

[3]http://www.juvenxu.com/2011/05/18/hudson-email-ext/