JIRA Advanced Mail Handler

6 Added by Daniele Raffo, last edited by Daniele Raffo on Dec 23, 2008

An extension of the default email handler that allows users to define a new issue's project, type, priority, reporter, components... in the email message.

Name	JIRA Advanced Mail Handler
Status	Stable
Version	2.3 (June 4, 2007)
Product Versions	JIRA 3.7
Author	Daniele Raffo, CERN
Price	Free
License	BSD (see license)
Homepage	JIRA Advanced Mail Handler
Issue Tracking	http://developer.atlassian.com/jira/browse/JMH
JavaDocs	http://www.crans.org/~raffo/projects/jmh/javadocs
Download Bytecode	advanced-mail-handler-2.3.zip advanced-mail-handler-2.3.tar.gz
Download Source	advanced-mail-handler-src-2.3.zip advanced-mail-handler-src-2.3.tar.gz

Description and Features

The JIRA Advanced Mail Handler is an extension of the standard JIRA email handler; it allows users to define many fields of the newly created issue (reporter, project key, type, priority, affected components, etc.) directly in the body of the email message. Hence, one of its notable features is that you do not need to repeat the whole process (opening a mail account on your mailserver, creating a mail account in JIRA, associate a handler service to the mail account and configure it) for each project you have in your JIRA installation; you just do the process once, setting a global mail account and handler service that will receive messages for all projects.

License

This software has been developed for CERN needs. It is available "as is" under BSD license and you are welcome to download and use it; however, **neither CERN nor the author (I) guarantees any support**. Please use the Comments section at the end of this wikipage to enter your question or feedback; note that I am automatically notified via email whenever a new comment is written.

This software is not under active development, so I do not plan to release any bugfix or feature request anytime soon. But if you are interested in helping out feel free to contact me.

Installation

To install the Advanced Mail Handler, follow these steps:

- Unzip the file archive and copy the compiled .class files into `WEB-INF/classes/cern/jira/emailhandler/` (you might need to create the required directories);
- Add the following snippet of code to `WEB-INF/classes/services/com/atlassian/jira/service/services/imap/imapservice.xml`

```
<value>
        <key>cern.jira.emailhandler.AdvancedCreateOrCommentHandler</key>
        <value>Advanced Mail Handler</value>
</value>
```

- Restart the JIRA server.

Configuration

Once the Handler is installed, you need to configure and activate it. This is done similarly as for the standard handler, as described in http://www.atlassian.com/software/jira/docs/latest/issue_creation_email.html: please read that page first, here I leave out some details and I describe only the steps that differ.
The procedure is as follows:

- Create an IMAP mail account, for instance `issues@yourcompany.com`. (The Handler probably works also on a POP account; however, I haven't tested it.)
- Add a new service: for Name, enter "Advanced Mail Handler"; for Class, select "Create issues from IMAP".
- Edit the newly created service: for Handler, select "Advanced Mail Handler"; for Handler Parameters, edit it as follows:
    - The `project` parameter is the key of the default project. An issue is assigned to this default project if the user did not specify any project key, or specified an inexistent project key, in the email message. This parameter is optional, but I recommend to set it.
    - The `issuetype` parameter is a number specifying the default issue type, and the association between numbers and issue types are explained in http://www.atlassian.com/software/jira/docs/latest/issue_creation_email.html#handler_params. An issue is assigned to this default type if the user did not specify the issue type in the email message. This parameter is mandatory.
    - The `jiraemail` parameter must be set to the address of the IMAP account, in our example `issues@yourcompany.com`. This parameter is mandatory.
    - If an alternate email address for `jiraemail` exists (for instance in the LDAP database of your company), the `jiraalias` parameter can be set to this alternate address. This parameter is optional.
    - The other parameters are the same as the ones defined for the default handler.

Now click the Update button and the service will be operative.

Usage

To comment an already existing issue, the usage is the same as the default handler: just mention the issue number in the `Subject:` of the email message. The body of the email will become the comment on the issue. You can also send attachments with your email, which will become attachments on the issue (if attachments have been enabled in JIRA).

On the other hand, the Advanced Mail Handler allows you to define several fields when creating an issue. This is done by adding one or more tags in the text of the `Subject:` header, in the format *#TAG-NAME=tag-value* or *#TAG-NAME=tag-value1,tag-value2,tag-value3...*. The tag must not contain any space, i.e. there is no space character around the equal sign or the commas.

⚠ Some user reported incompatibilities with newer versions of JIRA when sending HTML email messages. I recommend that you use plaintext format whenever possible (by the way, this is a good idea when writing emails in general...).

Project

You can specify the project in two ways:

- by addressing the message to *project-key* **<issues@yourcompany.com>**. This address can be put in the `To:`, `Cc:`, or `Bcc:` header at your choice.
- by adding the tag **#PROJECT=*project-key*** in the `Subject:` header. This overrides the project key in the address.

You always send the email to **issues@yourcompany.com** (or the address defined in the `jiraalias` parameter as well);

what changes is that in the first way you add the project key as the Full Name part of the address.
For your comfort, you can create an entry in your addressbook for each project you're working on, with the project key as the Full Name and issues@yourcompany.com (which will be the same for *all* projects) as the Email Address.
If you do not specify a valid project, the issue is created in the default project defined as the `project` parameter.

⚠

> Some mailclient (as Microsoft Outlook) does not give the user full control concerning the recipient address in the composing mail interface; for instance, if your company uses a LDAP database and issues@yourcompany.com has a record in it, these mailclients will not allow you to choose a Full Name for that email address and stubbornly put the Full Name defined in the database instead, no matter what you write. Also, some mailclient shows only the recipient Full Name and not the real email address, so it is difficult to tell who you are sending mail to. In this case you need to follow the second way and specify the project in the `Subject:` header.
> The mailclient might also convert issues@yourcompany.com to the alias defined in the LDAP database, e.g. Jira.Issues@yourcompany.com. In this case, in addition to setting `jiraemail` to issues@yourcompany.com, you must also set the optional `jiraalias` parameter to Jira.Issues@yourcompany.com.
> Nevertheless, to avoid all these hassles, my advice is to switch to a better mailclient – I personally recommend Thunderbird or SeaMonkey.

The `jiraemail` and `jiraalias` parameters have the purpose to ensure that a message was addressed to JIRA, so that spam does not go processed or, worse, ends up as an issue to the default project. If you see an `ERROR [cern.jira.emailhandler.MessageParser] Jira email address not found amongst recipients` in the logs, you specified these parameters incorrectly or you are sending the email message to the wrong address.

Summary

As with the standard handler, you write the summary of the issue in the `Subject:` of the email.

If you do not write any summary, the issue will be created with a summary text "(no summary)". However, it is recommended that you put a meaningful summary so all JIRA users can identify the issue.

ℹ The `Subject:` header can also contain several tags (explained further), which will not be included in the summary. These tags must be separated by spaces, and may be in any order.

⛔ You should not use special characters (letters with acute/grave accents, diaeresis, tilde, etc.) in the subject line. Please use only ASCII-7 characters.

Description

As with the standard handler, you write the full description of the issue in the body of the email.
You can also send attachments with your email, which will become attachments on the issue (if attachments have been enabled in JIRA).

Reporter

As with the standard handler, the reporter of the issue is identified as the sender of the email (you).

However, you might have more than one JIRA account associated with your email address, you might want to indicate another person as reporter, or you might be sending the email from another email account. In this case, you can explicitly specify the reporter by adding the tag **#REPORTER=***username* in the `Subject:` header.

Assignee

As with the standard handler, the first Cc'ed person that is also a valid Jira user and a valid assignee for the concerned project becomes the assignee.

As one person can have multiple JIRA usernames associated with his email account, you can explicitly set the assignee's username by adding the tag **#ASSIGNEE=*username*** in the `Subject:` header. This overrides the Cc'ed assignee.

If you do not specify any valid assignee, the issue is assigned to the Default Assignee.

Issue type

You can specify the issue type by adding one of the following tags to the `Subject`: **#BUG**, **#NEWFEATURE**, **#IMPROVEMENT**, **#TASK**, or **#SUBTASK**.

If you do not specify it, the issue will be of the type specified in the `issuetype` parameter.

Priority

You can specify the priority of the issue by adding one of the following tags to the `Subject`: **#BLOCKER**, **#CRITICAL**, **#MAJOR**, **#MINOR**, or **#TRIVIAL**.

This option was managed by the default handler by looking at the `X-Priority:` header of the email, but I implemented it as such because many mailclients do not permit to set custom headers easily.

Components

You can associate an issue to one or more components of the project by adding the tag **#COMPONENT=*component-name*** to the `Subject:`. To specify multiple components, separate the components by a comma, as in **#COMPONENT=*first-component-name,second-component-name,third-component-name***. Note that there is no space inside the whole tag.

ℹ If a component name contains any space, you must replace the space by two underscore characters. For instance, **#COMPONENT=Test__Component** associates the issue to Test Component.

If you specify no or invalid component, the issue gets associated by default to No Component.

Examples

✓ **Example I**

The minimal email message you can send is summarized below:

```
To: issues@yourcompany.com
Subject: #PROJECT=XYZ Race condition

A problem of race condition occurs when... etc.
```

This creates an issue of type specified in `issuetype` and of default priority (usually Major), on the project that has key XYZ, summarized as "Race condition".
You may also write the email in this format if your mailclient allows you to set the Full Name in the address:

```
To: XYZ <issues@yourcompany.com>
Subject: Race condition

A problem of race condition occurs when... etc.
```

Example II

Let's say you want to issue an improvement request of critical priority on the component core-alarms of project ABC, summarize it as "Alert user when a buffer overflow is reported", and assign it to John Hacker (John.Hacker@cern.ch).
You would usually Cc the message to John.Hacker@cern.ch; however, John Hacker has two JIRA accounts, `jhacker` and `admin`, associated with his email address, and you'd like to unambiguously assign the issue to the former.
Then you have to write an email as follows:

```
To: issues@yourcompany.com
Subject: #IMPROVEMENT #CRITICAL #PROJECT=ABC Alert user when a buff

It is crucial to show a visual warning to the user in case of a buf
```

Note that you can specify tags in any order and in any position, and they will *not* be included in the issue's summary.

Example III

To comment issue ABC-7, simply write the following email:

```
To: issues@yourcompany.com
Subject: ABC-7

My comment on this issue is... etc.
```

This is the usage of the standard JIRA email handler, and is left unmodified in the Advanced Mail Handler.

Troubleshooting

If you cannot get the Advanced Mail Handler to work, try the following:

- Read again this instruction page and the comments; it is easy to miss something the first time.
- Examine the JIRA logs to see what is going wrong.
- Check the incoming email in the IMAP account (where the Advanced Mail Handler fetches the mail from).
- Try to uninstall the Advanced Mail Handler and to use the default mail handler instead. If it fails too, chances are that some of your JIRA components (email server, services, project keys...) are misconfigured. In this case, escalate the problem and ask the general JIRA support for help.
- Using the Advanced Mail Handler, see if the problem persists when sending an email message in plaintext and with no attachments, as MIME parts may confuse the handler.

Tips and Tricks

If you prefer to have one email address for each project, like the standard email handler, just proceed as follows.
Create an IMAP email account for the first project, add a new service and associate the Advanced Mail Handler to it, as

explained in the Configuration section above. Specify the email address as the `jiraemail` parameter and the project as the `project` parameter. Repeat the process for each project.

Third-Party Patches

David Leinbach wrote a patch that allows the user to specify the due date (given as YYYY-MM-DD) and the original estimate of the issue. For instance: **#DUE=2008-04-14 #EST=1h**.

You'll have to apply the patch on the original source code, then recompile it. Note that the Advanced Mail Handler was compiled on Java 1.4.2_12 for compatibility issues with Tomcat. You can find a Windows version of the `patch` tool as a part of the GnuWin32 package. (These instructions are given here for your convenience; I haven't tried them.)

Labels

jira email handler

Comments (60)

Frank Metzger says: Jul 19, 2007

Using JIRA 3.10

Got an error while trying to load IMAP Mail..

```
2007-07-18 13:24:58,286 JiraQuartzScheduler_Worker-3 ERROR [service.services.mail.MailFetcherService] Throwable:
java.lang.NullPointerException
java.lang.NullPointerException
at cern.jira.emailhandler.MessageParser.emailAddressToRegex(MessageParser.java:188)
at cern.jira.emailhandler.MessageParser.parse(MessageParser.java:38)
at
cern.jira.emailhandler.AdvancedCreateOrCommentHandler.handleMessage(AdvancedCreateOrCommentHandler.java
at com.atlassian.jira.service.services.mail.MailFetcherService.run(MailFetcherService.java:190)
at com.atlassian.jira.service.JiraServiceContainerImpl.run(JiraServiceContainerImpl.java:67)
at com.atlassian.jira.service.ServiceRunner.execute(ServiceRunner.java:61)
at org.quartz.core.JobRunShell.run(JobRunShell.java:191)
at org.quartz.simpl.SimpleThreadPool$WorkerThread.run(SimpleThreadPool.java:516)
```

Here the settings:

Dienst bearbeiten: IMAP_ASS_TRANSFER

Anleitungen:
Geben Sie die Diensteigenschaften unten in Form von Textwerten ein. Leere Felder werden bei der Dienstinitialisierung auf NULL gesetzt.

Handler:	Advanced Mail HandlerErstellungs- oder KommentarhandlerVorgangshandler erstellenNicht angegebener KommentarhandlerVollständiger KommentarhandlerCVS-Protokoll-Handler (veraltet)
Handler-Parameter:	
Verwendet SSL:	Keine SSLSSL (POP/S)
E-Mail weiterleiten:	
Server:	ASS_TRANSFER
Ordnername:	Name des Ordners eingeben, aus dem gelesen werden soll. Leer lassen für Standard (INBOX).
Sie können den Verzögerungszeitraum des Dienstes anpassen. Beachten Sie, dass der Dienst nach der Umstellung der Verzögerung neu gestartet wird.	
Verzögerung:	Verzögerung in Minuten

Any Ideas?

Daniele Raffo says: Jul 24, 2007

The error is due to the fact that the `jiraemail` parameter was not specified in Handler Parameters.

mameha says: Sep 25, 2007

To get this to work for POP, I simply add the snippet of code to the [blah]/pop/popservices.xml? Or is there something else required?

Daniele Raffo says: Oct 03, 2007

You may try; I haven't tested it but I don't see why it should not work.

Of course you also need to set up a POP mail account, and the Jira service must be of class "Create issues from POP".

Tennis Smith says: Oct 27, 2007

Is there any way to add a fisheye link to an issue using this?

Daniele Raffo says: Oct 28, 2007

I haven't been working with Fisheye recently, but if I do remember well, what you want is done automatically by Jira/Fisheye basing on the tags in the source code repository (if you mention an issue number when doing a CVS commit, then that issue will automatically be linked to your changes in the source).

Tennis Smith says: Oct 28, 2007

That's what I will normally do. This is a little different. I'm using jira to track test case implementation. There are situations where I need to specify what test(s) are being implemented by a given function. To do that, I put a test name in the function's comments. Once done, I would like to point to the specific line in the code where a test is implemented. For example:

```
/*
* This is implementing testcase issue
* TEST-222
*/
```

I would then like to put something like this inside the TEST-222
issue comments:

http://fisheye/browse/foo.c?r=1.14#l101

I'd like to add these comments via your mail handler. 🙂

Peter Moore says: Oct 28, 2007

Hi Tennis,

JIRA 3.12 will have an Application Linker plugin that will simplify linking to FishEye. i.e. you will be able to define a pattern that gets converted to a deep link. For the moment the best idea is just to add a comment with the FishEye url in it.

I've raised FISH-107 to include content results in the FishEye tab. At the moment only content on Head/Trunk/Mainline is indexed though, so you probably still want to add permalinks as comments.

Cheers,
Pete

Tennis Smith says: Oct 28, 2007

Thanks Peter. That will certainly help.

Making all the links manually will be a problem however. I'm about 1/4 of the way through my project and I already have just under 1000 links to add.

I'm thinking of try to do this via csv imports for now.

Thanks again
-Tennis

Tennis Smith says: Oct 29, 2007

The handler appears to split any comment string with a newline at about column 70 or so. Is that expected? Can that value be changed? My preference would be for a newline at about column 120.

Thanks,
-T

Daniele Raffo says: Nov 20, 2007

It is probably your mailclient that wraps text at column 70. Try changing the settings of your mailclient.

Ryan says: Oct 31, 2007

Is there a way to pass custom field values in through the mail? For example, if I have a custom field: customfield_10005FieldArea with a value of "test value", can I pass that like I do for the other built in values like #ASSIGNEE=*username ?*

Daniele Raffo says: Dec 23

No, the parser recognizes only the tags listed above. Sorry.

mameha says: Nov 20, 2007

If you could add that function it would be very powerful.

I need to import issues form an external system somehow and populate custom fields. I tried using the excel POP parser extension (http://confluence.atlassian.com/display/JIRAEXT/Excel+POP+Parser+Handler) but I can't get it to work.

CYTEXONE says: Dec 31, 2007

How difficult would it be to allow the parser to recognize custom fields? It would be so awesome.

Would you take donations to make this happen?

Daniele Raffo says: Jan 13, 2008

Thank you, but actually I am busy with other projects and I have no time to work on the Advanced Mail Handler. I'm sorry.

Modifying the source code of the parser to recognize a custom field e.g. "customfield_10005FieldArea" would be quite straightforward: everything is done in MessageParser.parseSubject() and you would just need to add a new parsing rule, basing on the existing code.
However, you also need to manage a MutableIssue object to include your custom field and its value, and this is outside the scope of my parser.

Limwell Alim says: Jun 11, 2008

hi mate,

is there a possibility that this feature be on Advance mail handler? or could you suggest any workaround on how to do this. Thanks 🙂

Daniele Raffo says: Jun 11, 2008

Hi,

Well, I see no workaround for this apart from Java coding...

Matthew Lieder says: Nov 28, 2007

Are the downloads valid ZIP files? I've been unable to open either of the files so far with any ZIP extractor I've tried...

Daniele Raffo says: Jan 13, 2008

Yes, they are. Please try again – another user reported the same problem and eventually he was able to open them.
Anyway, I have now packaged the same files in gzipped tar archives, as an alternative to zipfiles.

Silvan Mühlemann says: Jan 16, 2008

The default behaviour of the mail handler with cc'ed adresses is them to become the assignee.

Is there a way to put them as *watcher* instead of as *assignee*?

Daniele Raffo says: Jan 20, 2008

No, this feature is not implemented.

wai yan says: Jan 18, 2008

i cant extend this feature. Is there anybody who has successfully can use this feature? please help me how to make it work. I set the parameter correctly. But it is not working.

usessl:	No SSL
popserver:	testing
handler.params:	project=JIRA, jiraemail=jira@tesing.com
handler:	Advanced Mail Handler

Daniele Raffo says: Jan 22, 2008

Hello Wai,

1) I have not tested the handler with POP – just with IMAP.
2) Check that you have specified the correct name for your POP server.
3) Check that the mail account JIRA fetches messages from (jira@tesing.com) is correct and accessible.
4) Read the JIRA logs to know exactly why it is not working.

Sophie Wong says: Feb 04, 2008

Is there a way to update an existing issue using this handler? I would like to reassign an issue or even close an issue via an email.

Daniele Raffo says: Feb 14, 2008

No, you can just create or comment an issue – like the original email handler allowed to do.

Daniele Raffo says: Mar 07, 2008

Received from an user and posted here (with her consent) for the benefit of the other users. My reply is below.

We have two Helpdesks, IT and SYSTEMS. Issues are sent to the specific mailbox and the technician solves the problem from the mailbox.

I want to create the issues in each of the mailboxes as issues in separate projects. Jira, however, can only handle one mailbox. I know I can use your Advanced Handler to create issues in different projects.

However, the issues in the two Helpdesks are confidential per Helpdesk, thus the messages cannot be merged into one mailbox for the issues to be created in Jira.

I have overcome that by creating a rule in mailbox IT and another rule in mailbox SYSTEMS to forward the mail to a Jira mailbox (not accessible by the users) with the text attach to the subject as #PROJECT=SYSTEMS or #PROJECT=IT, depending where the mail comes from.

Unfortunately our mail client add the text at the end of the subject, for ex: FWD:Printer Problem (#PROJECT=IT).

The Advanced Email Handler only reads the text if it is first in the subject (for example: #PROJECT=IT Printer problem) and then adds it to the correct project.

Is there a way that the Advanced Email Handler can pick up the #PROJECT, even if it is not the first text in the subject or do you have another solution for me? I unfortunately cannot change the email subject text.

To solve your problem you can create two email accounts (one for IT and one for SYSTEMS) in Jira, and two different Advanced Mail Handler services. Each service will fetch only email collected in its own account. Remember to set the default project for each account (i.e. IT and SYSTEM respectively) so you do not need to specify it in the Subject.

Note that it is not true that the tags need to be first in the Subject text. You can specify tags in any order, so "#PROJECT=IT Printer problem" and "Printer problem #PROJECT=IT" and even "Printer #PROJECT=IT problem" are equivalent and will all work. The Advanced Mail Handler cannot find the project tag in your example because it is surrounded by parenthesis.

Neeraj Jhanji says: Mar 31, 2008

Does this plugin work on JIRA 3.12.2? We installed it but are having problems when trying to configure it.

Daniele Raffo says: Apr 07, 2008

Hi Neeraj,

I developed and tested the handler under Jira v3.7, I don't know about compatibility issues with v3.12. Maybe someone can share its experience.

Which problem do you have exactly?

Neeraj Jhanji says: Apr 04, 2008

Hello Daniele,

When we tried to install the Advanced Mail Handler on JIRA 3.12.2, it seemed to interfere with JIRA's built-in Create/Comment Handler. After installing it (and making the necessary changes to the XML files), it was no longer possible to add any new services to JIRA and began to throw a nullpointer exception: "Error adding service: java.lang.NullPointerException". The problem was fixed by removing the Advanced Mail Handler.

We would like to use if you can make sure it works on JIRA 3.12.2 and an in unison with the built-in email handler. Could you please update your code for that?

best regards,

Neeraj

Daniele Raffo says: Apr 07, 2008

This is strange. Have you tried to add the Advanced Mail Handler as the last service? Or to use just one Handler?

I am busy with other projects, therefore I cannot currently plan to do further development/testing on the Advanced Mail Handler. However, the full source code is available for those who want to take the challenge. That's all I can suggest at the moment. Sorry to disappoint you.

Sophie Wong says: Apr 03, 2008

I have been using this plugin and find it extremely useful. Thanks for creating it!

One problem I am having is that the plugin does not parse a custom issue type and priority I created. When I try to use the new issue type or priority, the plugin treats it as part of the summary. Is there a configuration file that needs to be modified?

Daniele Raffo says: Apr 04, 2008

Hi Sophie,

The Advanced Mail Handler is able to recognize only the default values for issue type and priority. If you are able to do some Java programming you can modify the handler's source code to include your customization; it is really simple.
If someday I find the time to release a new version of the handler I will rewrite the parser so users will be able to easily customize it via a text configuration file.

Thanks for the feedback 🙂

Daniele Raffo says: Apr 23, 2008

Received from an user, my reply is below.

Basically, I would like to extend your work to read tags from the body part of an e-mail. Based on your experience, do you think it is something worth trying?

Sure it is worth trying. You can write a parser to analyze the email body as well as the original parser of the Advanced Mail Handler analyzes the Subject. My advice is that you reserve the first line(s) of the email body to store the tags. Just remember to strip out the tags, because the email body is used for the issue description text and you probably want the tags not to be included in the description.

Paul Prendergast says: May 16, 2008

I am having problems with support custom issue types.

The handlers I have setup to support all my custom issue types and noticed it cannot support an issuetype name with multiple words.

EX) "Business Process Change" for the filter on the subject line i would use #Business Process Change....but did not work.

I have tried using the double unscore like in the notes for supporting components and it still did not work

I also tried using the issuetypeid like #17 that you can find in administrator issuetype section when rolling the mouse over edit link and looking at the bottom of the browser.

I am running out of options can you help?

Daniele Raffo says: May 19, 2008

Hi Paul,

The Handler recognizes only the following issue types: #BUG, #NEWFEATURE, #IMPROVEMENT, #TASK, and #SUBTASK. If you would like to include your custom issue types, you would need to modify and recompile the source code.

Steve Lane says: May 31, 2008

Hi Daniele -- this looks like a great tool. I am trying to install it in Jira 3.12, but the file WEB-INF/classes/services/com/atlassian/jira/service/services/imap/imapserv seems not to exist. So the service is never really recognized. Do I need to create that file?

Thanks for any insight.

-- SGL{}

Daniele Raffo says: Jun 05, 2008

This is strange as it's the JIRA classpath. Have you tried to ask for support in the JIRA forum?

Tennis Smith says: Jul 01, 2008

Is there still no way to close or resolve an issue from this interface?

-Tennis

Daniele Raffo says: Jul 03, 2008

Not actually, sorry. If some Java developer would like to implement this feature I'd be happy to host the patch and provide a link on this page (as for Leinbach's patch for Due Date and Estimate).

Christina m. says: Jul 14, 2008

I know this is an extremely basic question, but could you tell me exactly where in the code that the snippit should be added? Thanks.

Daniele Raffo says: Jul 15, 2008

It makes no difference. You may insert it at the end of the file.

Joerg Schütte says: Jul 30, 2008

Hi,

Thanks for implementing this nice feature. Unfortunately I do have the problem that I see an error message in the Logs if the Jira address is not in the TO: field but in the CC: field.

Logs say:

```
2008-07-30 09:13:58,256 JiraQuartzScheduler_Worker-1 WARN
[service.util.handler.AbstractCommentHandler] Jack Black does not have permission to comment on an
issue in project: 10211
```

The User has full permissions to create and comment all tickets and I tried to forward it directly to my Jira address and now it is in the system.

BTW: I am using Jira 3.11!

Thanks for looking into this.

J:org

Daniele Raffo says: Aug 21, 2008

This is strange as the log reports a problem with user permissions but, as you're telling me, this happens only when JIRA is in Cc:.

Note that I wrote the Handler with JIRA 3.7. Perhaps the way JIRA manages mail messages has changed since. Have you tried asking for support in the JIRA forum?

As a workaround, you can put the JIRA address in the To: field.

Andy Brook says: Sep 10, 2008

Hey Daniele,
I have some email handler code that does some fancy LDAP lookup of alternate email addresses and also supports (did support) the Confluence MailFormNG forms for inbound property setting much like you are doing here. I was thinking about setting up a project for it, but then again, wonder if its worthwhile sticking the code into this project. What do you think? I can get the SNV repo and M2 build stuff setup...

Daniele Raffo says: Sep 18

Hi Andy,

Are you thinking of merging LDAP lookup functionality to the Advanced Email Handler? There are some issues concerning the use a LDAP database, as written in the instructions (basically some dumb mailclient replaces the email addresses when sending, and this confuses the parser). Therefore I don't know how much this new feature would be useful.

Of course you are very welcome to experiment with the Advanced Email Handler code; if you set up a project based on it, let me know so I can link to it on this wikipage.

Andy Brook says: Nov 04

Hi Daniele, just re-reading this, yes, I had the problem on my network, inbound email doesn't work for 60% of users. For me (tm) in Active Directory land, there is a default mail account that users are registered with in Confluence/Jira, but their from: email tends to be an 'alias'. The code I wrote scans the aliases to find the related primary email address in order to find the user, tada! Of course this won't work for *everyone* so there is the option of implementing your own AlternateIDProvider.

Rather than just add something here, I created a framework (Jira Extendable Mail Handler) to allow easier addition and support of *mulitple* 'mail processors' within a single handler. For example If I have one mailbox and some users like to use this format whereas others want a different one, how to support? Well, in order to say support this format, all we need is one class that maps from email (in this case subject parsing) to a common map of keys to values, re-using common issue creation logic. Its gotta be such a pain for everyone wanting to support different inbound email formats and having to go through the learning curve of how to create all the various bits of a Jira issue. Fancy creating a new impl of your format? If not, feel free to rip anything useful 🙂

Erich Weihrauch says: Oct 21

Awesome plugin!

Daniele Raffo says: Oct 22

Thank you. 🙂

Christine ROLLAND says: Jan 20

Hi Daniele,

Due to audit purposes, I must include the original request (email) to the jira.

Is it possible to do it with your plugin ?

Thanks

Daniele Raffo says: Jan 21

Well, yes, and the simplest way to do that is to use your mailclient to Forward the email request to the JIRA mailbox.

Carmon Racheli says: Jan 21

Hi Daniele,

Thanks for your Advanced Mail Handler.

Do you have any idea what it the problem?

I defined service (default without your advanced) using IMAP and it works. The email account was also ok and issues inserted to jira.

I used the same IMAP server and the same email account - no issues inserted to JIRA (I have jiraemail in handler.params*)* and get the following error in JIRA log:

```
2009-01-21 18:37:15,293 JiraQuartzScheduler_Worker-2 ERROR [service.services.mail.MailFetcherService]
Error connecting to host 'webmail.ihostexchange.net' as user support@xxx.com via protocol 'imap':
javax.mail.MessagingException: Connection timed out: connect;
  nested exception is:
 java.net.ConnectException: Connection timed out: connect
javax.mail.MessagingException: Connection timed out: connect;
  nested exception is:
 java.net.ConnectException: Connection timed out: connect
 at com.sun.mail.imap.IMAPStore.protocolConnect(IMAPStore.java:441)
 at javax.mail.Service.connect(Service.java:233)
 at javax.mail.Service.connect(Service.java:134)
 at com.atlassian.jira.service.services.mail.MailFetcherService.run(MailFetcherService.java:149)
 at com.atlassian.jira.service.JiraServiceContainerImpl.run(JiraServiceContainerImpl.java:67)
 at com.atlassian.jira.service.ServiceRunner.execute(ServiceRunner.java:61)
 at org.quartz.core.JobRunShell.run(JobRunShell.java:191)
 at org.quartz.simpl.SimpleThreadPool$WorkerThread.run(SimpleThreadPool.java:516)
```

Thanks, Racheli

Yoz Grahame says: Jan 27

We've just upgraded and substantially reconfigured our JIRA install, and we're now seeing these errors continually in the logs:

```
2009-01-27 13:47:06,079 JiraQuartzScheduler_Worker-3 ERROR [service.services.mail.MailFetcherService]
CR Project Service[10010]: Exception: nulljava.lang.NullPointerException        at
cern.jira.emailhandler.MessageParser.parseSubject(MessageParser.java:74)        at
cern.jira.emailhandler.MessageParser.parse(MessageParser.java:59)        at
cern.jira.emailhandler.AdvancedCreateOrCommentHandler.handleMessage(AdvancedCreateOrCommentHandler.ja
at com.atlassian.jira.service.services.mail.MailFetcherService.run(MailFetcherService.java:189)        at
com.atlassian.jira.service.JiraServiceContainerImpl.run(JiraServiceContainerImpl.java:56)        at
com.atlassian.jira.service.ServiceRunner.execute(ServiceRunner.java:61)        at
org.quartz.core.JobRunShell.run(JobRunShell.java:191)        at
org.quartz.simpl.SimpleThreadPool$WorkerThread.run(SimpleThreadPool.java:516)
```

Any ideas? It's driving us nuts, and we're going to have to disable the Advanced Mail Handler if we can't solve it (which would make us sad as it's a fantastic plugin!)

Thanks for any help you can provide!

Isaac says: Feb 10

Heya Daniele. We recently upgraded to Jira Version: 3.13.2-#335. After upgrading to this version, any time we send an email to our queue, it seems to loop and creates hundreds of tickets. I'm not noticing any errors in the log files after this. Any idea what this could be or how I could resolve it? It worked perfectly fine before the upgrade.

Thanks!

- Isaac

Alexander Weiss says: Feb 11

Hi,

there is indeed an issue in the combination of the current advanced email handler and the new JIRA version 3.13.2#335. We have exactly the same problem: The email will be imported as JIRA ticket but the email will not be deleted in the inbox afterwards.

Any Idea?

So we stuck in an JIRA update trap since we are unsing the advanded email handler extensively

Thanks for help!

Alexander

Daniele Raffo says: Feb 11

Hi Isaac and Alexander,

please have a look at bugreport JMH-4 at http://developer.atlassian.com/jira/browse/JMH-4 . This problem was observed when sending email in HTML format. The bugreport page also contains the patches (developed by users) to fix the problem.

Alexander Weiss says: Feb 11

Hi Daniele,

thanks for the quick repsonse. But unfortunately this is not the solution to our issue since the emails we are using are plain/text and not html. Referencing to the issue JHM-4 there is a error-log mentioned but we don't get any error message as Isaac doesn't get any one.

Any idead?

Cheers,

Alexander

Carmon Racheli says: Feb 16

Hi,

Sending an email to JIRA (IMAP) worked but sending an email using the advanced email handler with the same definition of service fails.

Any idea ?

Haramohan Nanda says: Feb 26

Hi

We have "**#QUESTION #MAJOR**" in our subject line in the mail and also we have created Issue type named Question in the jira,But when the mail handler reads the message it creates a **trival** type **bug** in the jira.

All the rest in the header like "#COMPONENT=asdzxc #ASSIGNEE=zxyuser" are working fine.

Also can you suggest us that how can we add an user to watcher list these mail handler.

Thanks for any help you can provide!

Haramohan

Daniele Raffo says: Mar 11

Hi Haramohan,

You should use the default issue types, or otherwise modify the code of the Advanced Mail Handler to reflect the changes in your custom issue types. The Handler does not recognize the #QUESTION value and sets the issue to the default type -- BUG.

The issue Watcher tag is not implemented in the Handler.

Printed by Atlassian Confluence 2.10.2-rc1, the Enterprise Wiki.