***Parameters for jira-custom-advanced-email-handler***

For all handlers :

* fspack=true/**false**

set the following options as default :

* + - createUsers=true
    - referenceAttachments=true
    - htmlFirst=true
    - forceProject=true
    - ccAssignee=false
    - stripQuotes=true
  + If those parameters are also set in the configuration string, they will replace the values in fspack.
* reporterusername=<username>

default reporter username

* catchemail=<Email Address>

The message will be handled only if the catchemail address is found in the To: Cc: or Bcc: fields

* bulk=ignore/forward/delete

handles the messages where the Precedence: field is set to bulk, or Auto-Submitted is set to no in the header. This option tells the handler to ignore the email, forward it to the email address set in the forwardemail parameter, or delete the email permanently.

* referenceattachments=true/**false**

If this parameter is set to true, all message’s attachments are referenced in the description (when the issue is created) or in the comment (when a comment is added).

* createusers=true/**false**

Tells the handler to create new user accounts from any received email messages whose From: field contains an address that doesn’t match one associated with an existing JIRA user account. This option is not compatible with the reporterusername option. Createusers will take precedence on reporterusername.

* notifyusers=true/**false**

if set to true, users whose accounts have been created by the createusers option. This option only works if the createusers option is set.

* adduserstogroup=<groupname>

if this parameter is set, the handler will remove the newly created users from the default groups, and add them to <groupname>.

* htmlfirst=true/**false**

if set to true, the handler will try to retrieve the html part of the email first. Otherwise, the text part will be used.force

* forceproject=true/**false**

if set to true, the handler will always create the issue in the project specified in the project parameter. If an issue (SUP-1 for example, SUP being the default project) has been moved to another project (DEV-1) and a message references SUP-1 or DEV-1 in the subject, the message won’t be used as a comment for DEV-1, but will cause the creation of a new ticket (SUP-2).

* fingerprint=accept/**forward**/ignore
  + Defines the fingerprint policy. See com.atlassian.jira.service.util.handler.MessageHandler.
* ccField=<field name>
  + If this field is set, all email addresses contained in the To: or Cc: fields will be created if needed, and added to the customfield in the parameter value. This must be a Multi User Picker.

For AdvancedCreateIssueHandler (this handler is also called when using AdvancedCreateIssueHandler and an issue doesn’t exist) :

* project=<Project Key>

Default project. Issues will be created in this project if they don’t already exist, or if the forceproject parameter is set to true.

* issuetype=<IssueType Id>

Default Issue Type. Issues will be assigned to this issue type if they don’t already exist.

* ccAssignee=**true**/false

If true, the first cc User becomes the assignee.

* component=<Component Name>

default component for the new issue

* excludeaddress=<email Address>

don’t create issue from messages sent by this excluded email address

For AdvancedCreateOrCommentHandler :

* project=<Project Key>

Default project. Issues will be created in this project if they don’t already exist, or if the forceproject parameter is set to true.

* issuetype=<IssueType Id>

Default Issue Type. Issues will be assigned to this issue type if they don’t already exist.

* stripquotes=true/**false**

remove quoted text from from an email message's body (e.g. from previous email replies) before the body's content is added to the JIRA issue's comment

* jiraemail=<Email Address>

Mail address the handler is listening on. When handling the address list, don’t care of this address to create new users.

* customfield=<field id :value to set>

name/value of the customfield to set when handling the message. The format is « 10000:text ».

* jiraalias=<Email Address>

Same purpose of jiraemail, if an alias exists.

* whitelist=<Email Address>

if the sender is unknown to JIRA, allows messages sent by the whitelist (To Be Checked)

* subjectregexp=<regular expression>

Check if the subject contains a pattern and replace it with the subjectreplace parameter

* subjectreplace=<value to replace> (used with subjectregexp)