Actions

Filters (tests)

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
<th>Example</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>@username</td>
<td>Selects messages sent by a particular user</td>
<td>@rhirsch</td>
<td></td>
</tr>
<tr>
<td>any</td>
<td>Selects all messages regardless of their characteristics</td>
<td>any</td>
<td></td>
</tr>
<tr>
<td>/regex/</td>
<td>Selects messages if body matches a regular expression</td>
<td>/exception/</td>
<td></td>
</tr>
<tr>
<td>&quot;string&quot;</td>
<td>Selects messages if body contains the case insensitive string</td>
<td>&quot;sales order&quot;</td>
<td></td>
</tr>
<tr>
<td>#tag</td>
<td>Selects the messages containing the tag.</td>
<td>#project gumbo</td>
<td>Note the tag could be in the message as a hash code or created by the user directly as a tag</td>
</tr>
<tr>
<td>to = @username</td>
<td>Selects the messages that were sent to a particular user</td>
<td>to = @rhirsch</td>
<td></td>
</tr>
<tr>
<td>to = (@username1, @username2)</td>
<td>Selects the messages that were sent to multiple users.</td>
<td>to = (@rhirsch, @m)</td>
<td>List should be comma separated</td>
</tr>
<tr>
<td>to &lt;&gt; @username</td>
<td>Selects the messages that were not sent to a particular user</td>
<td>to &lt;&gt; @rhirsch</td>
<td></td>
</tr>
<tr>
<td>not(exp)</td>
<td>Selects the messages in which the expression evaluates to false</td>
<td>not(&quot;sales order&quot;)</td>
<td></td>
</tr>
<tr>
<td>exp1</td>
<td>exp2</td>
<td>Selects the messages in which either expression is true</td>
<td>&quot;sales order&quot;</td>
</tr>
<tr>
<td>exp1 &amp; exp2</td>
<td>Selects the messages in which both expressions are true</td>
<td>&quot;sales order&quot; &amp; &quot;profit&quot;</td>
<td></td>
</tr>
<tr>
<td>number%</td>
<td>Returns messages based on the percentage.</td>
<td>44%</td>
<td>For example, the 45% will be true 45% of the time.</td>
</tr>
</tbody>
</table>

Hooks (messages generated on system events)

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>login</td>
<td>Selects the messages generated on user login.</td>
<td></td>
</tr>
<tr>
<td>follow</td>
<td>Selects the messages generated when a user follows you.</td>
<td>%f expands to follower nickname</td>
</tr>
<tr>
<td>unfollow</td>
<td>Selects the messages generated when a user unfollows you.</td>
<td>%f expands to ex-follower nickname</td>
</tr>
<tr>
<td>profile</td>
<td>Selects the messages generated when your profile is changed.</td>
<td></td>
</tr>
<tr>
<td>every number minutes</td>
<td>Generates messages at regular intervals and selects them.</td>
<td>The message is not saved- too noisy, plus two identical tests never refer to the same event</td>
</tr>
</tbody>
</table>

User, followers' and public timelines
Messages generated by events can be used in the actions test expression, since the event cause is attached to them. Messages generated by events appear by default in the user's timeline, but not in follower's timeline. The message can be resent to followers using the existing resend action, or suppressed from user's own timeline by means of the filter action. Any and all saved messages appear in the public timeline (this could be addressed by these messages being private by default in the future).

Actions (performances)

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
<th>Example</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>filter</td>
<td>Filter the message out and don't put it in your mailbox</td>
<td>filter</td>
<td></td>
</tr>
<tr>
<td>resend</td>
<td>Resend the message to all your followers regardless of their characteristics</td>
<td>resend</td>
<td>It will only be put in your followers timeline if it is not already there.</td>
</tr>
</tbody>
</table>
http://server
Does an HTTP(S) POST of the message on a server with certain headers set

http://user:password@identi.ca/api/statuses/update.xml
status=%s

Enter HTTP POST data on a new line; %s expands to original message, %u expands to user nickname, %i expands to image URL, %w expands to user’s full (mnemonic: w hole) name (first + last)

mailto: email
Send the body of the message to the email recipient

mailto:wille.coyote@acme.com
For custom mails, enter mail text on a new line; same expansions apply as for HTTP POST

atom://
Posts the new entries from Atom/RSS feed

atom:http://twitter.com/statuses/user_timeline/esmeproject.atom
Bots can also post to Apache ESME, but polling from Apache ESME is more secure and robust

scala
Interprets the text of a message as Scala code and returns back the result

action which interprets the text of a message as Scala code and returns back the result
Details here

Examples for assembling tests and actions

<table>
<thead>
<tr>
<th>Test</th>
<th>Action</th>
<th>Description</th>
<th>Similar to</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>mailto: mail</td>
<td>Sends mail when followed by any user</td>
<td>Twitter sends messages on follow</td>
</tr>
<tr>
<td>unfollow</td>
<td>mailto: mail</td>
<td>Sends mail when unfollowed by a user</td>
<td>Qwitter helps track who is unfollowing you</td>
</tr>
<tr>
<td>profile</td>
<td>http:// server</td>
<td>Sends HTTP POST when the profile is changed</td>
<td>@chinpisin on Twitter tracks your avatar change to chinposin.com</td>
</tr>
<tr>
<td>every 5 mins</td>
<td>rss:<a href="http://blog.esme.us/feed/rss">http://blog.esme.us/feed/rss</a></td>
<td>Updates new posts from the ESME blog</td>
<td>twitterfeed posts your blog titles to Twitter</td>
</tr>
<tr>
<td>login</td>
<td>resend</td>
<td>Notifies your followers when you log in</td>
<td>Yammer posts your blog titles to Twitter</td>
</tr>
<tr>
<td>#twitter &amp; @me</td>
<td><a href="http://user:password@twitter.com/statuses/update.xml">http://user:password@twitter.com/statuses/update.xml</a> status=%s</td>
<td>Posts all messages tagged #twitter to Twitter</td>
<td>Ident.ca integrates with Twitter by reposting messages</td>
</tr>
<tr>
<td>#bot</td>
<td><a href="http://localhost:8080/api/send_msg">http://localhost:8080/api/send_msg</a> token=XXX&amp;message=%u said: %s</td>
<td>Reposts customized message to the same ESME instance, possibly with a different user’s credentials</td>
<td>twitterbots helps you retweet automatically</td>
</tr>
</tbody>
</table>

Too permissive action tests
Be careful, tests for the actions might apply to more messages than you think. For example, the tag filter applies not only to your messages, but also ones from people you’re following. In the action above which posts to Twitter, if you forget to restrict the author, you might flood the twitter timeline with messages you didn’t intend to post.

Additional Links
There are many lists of ad-hoc Twitter third-party tools, some of which might be assembled in ESME using actions.

- http://tweeternet.com/
- http://www.squidoo.com/twitterextensions