rsyslog[1]

The rocket-fast system for log processing

Tag Archives: template

We have just released 8.3.2 of the v8-devel branch.

This is primarily a bug-fixing release, but it also adds the ability to extract parts of a timestamp via the property replacer and adds support for per-listener imrelp rulesets (thanks to bobthesecurityguy for the patch!).

ChangeLog:

http://www.rsyslog.com/changelog-for-8-3-2-v8-devel/[2]

Download:

http://www.rsyslog.com/download-v8-devel/[3]

As always, feedback is appreciated.

Best regards,
Florian Riedl

Tags: 8.3.2[4], bugfix[5], devel[6], imrelp[7], mmexternal[8], release[9], rsyslog[10], template[11], v8[12]
Version 8.3.2 [v8-devel] 2014-05-02

- new template options for date extraction:
  - month
  - minute
  - second
  - tzoffshour

- tzoffsmin
- tzoffsdirection
- wdayname

  For string templates, these are property options and they are prefixed with "date-" (e.g. "date-year", "date-month", …)

  see also: https://github.com/rsyslog/rsyslog/issues/65[13]
- bugfix: mmexternal remove framing char before processing JSON reply
  This did not have any real bad effects, but caused unnecessary processing, as empty replies were not properly detected. Otherwise, the bug was not noticible from the user's PoV.
- bugfix: mmexternal segfault due to invalid free in non-json input mode

  closes: https://github.com/rsyslog/rsyslog/issues/70[14]
- bugfix: mmexternal segfault when external plugin sent invalid reply … or no reply at all. This happened if the reply was imporper JSON. Now, we emit an error message in those cases.

  see also: https://github.com/rsyslog/rsyslog/issues/69[15]
- bugfix: mmexternal did potentially pass incomplete data to restarted external plugin
  This could happen if EPIPE was returned "too late", in which case the beginning of the data could be lost.
- bugfix: mmexternal did not properly process messages over 4KiB
  The data to be passed to the external plugin was truncated after 4KiB.

  see: https://github.com/rsyslog/rsyslog/issues/64[16]
- imrelp: added support for per-listener ruleset and inputname

  see: https://github.com/rsyslog/rsyslog/pull/63[17]
  Thanks to bobthesecurityguy github user for the patch

Tags: 8.3.2[18], bugfix[19], Changelog[20], devel[21], imrelp[22], mmexternal[23], rsyslog[24], template[25], v8[26]

If you are using a lot of filters and templates in rsyslog, this can not only be affecting the performance drastically, but it is also a hassle to set up all the different actions and templates. It is always worthy to check, if there isn't a shortcut somewhere, which might not only save you time for creating the configuration, but also make it much simpler in the end to keep track of all the actions.

In our example, we have several programnames. The log messages should be sorted by programname and then be stored in a specific file and be sorted by host. After storing the log messages, the message should be discarded, so it won't be processed by the following filters, thus saving otherwise wasted processing time. **This example is applicable to rsyslog v7.2.x and above.**

Here are some sample config lines.

```
## apache_access
template(name="DailyPerHost_apache_access" type="string" string="/syslog/%FROMHOST%/apache_access.log")
:programname, isequal, "apache_access" { action(type="omfile" DynaFile="DailyPerHost_apache_access") stop }

## apache_error
template(name="DailyPerHost_apache_error" type="string" string="/syslog/%FROMHOST%/apache_error.log")
:programname, isequal, "apache_error" { action(type="omfile" DynaFile="DailyPerHost_apache_error") stop }
## mysql
template(name="DailyPerHost_mysql" type="string" string="/syslog/%FROMHOST%/mysql.log")
:programname, isequal, "mysql" { action(type="omfile" DynaFile="DailyPerHost_mysql") stop }

## php
template(name="DailyPerHost_php" type="string" string="/syslog/%FROMHOST%/php.log")
:programname, isequal, "php" { action(type="omfile" DynaFile="DailyPerHost_php") stop }
```

These are some basic services, which are often run together. Please note, that these are just a few examples. As you can see here, the *template* is created first. It is given a name, type and format. Templates of type string are usually used for file names. Here the log messages get stored in the folder /syslog a subfolder for the host where the message occured and then a filename which reflects the type of message that occured.

The second line holds the actions. First you see the property based filter (*programname*) and the condition. After that the actions get chained with the curly braces. The log messages where the filter evaluates to true get stored in a file. The filename and path is generated dynamically with the *DynaFile* parameter. Through this, the above written *template* will be used to generate the path and filename. The second action is represented by *stop*. Please note that this is case sensitive. Basically, *stop* means to stop the message processing. No further processing of the message will take place.

If we look closely at the sample config lines, we see, that the filter condition is basically always the same. It will always filter the *programname* property for a certain value. This is a predestinated case for using an array for simplification. We can use the property *programname* in the file template as well and filter an array of values. This will greatly save the overhead for all the seperate filter, not only in the configuration, but also in processing the messages.

```
template(name="DailyPerHost_app" type="string" string="/syslog/%FROMHOST%/%programname%.log")
if $programname == ["apache_access",
                    "apache_error",
                    "mysql",
                    "php"]
```

```
then {
action(type="omfile" DynaFile="DailyPerHost_app")
stop
}
```

Again, we first create the *template*. Please note the difference in the filename where the hardcoded text has been replaced by the property *programname*. In the next lines, we see the filter and the *array* of values. This is just to reflect the example. Virtually, the array can have near-infinite values. The filter is also a common if/then construct. After the then we see our chain of commands. First the action which writes the log messages into a file where the filename is created by the above template and then a stop as second action.

This case is applicable in many forms. It is also most useful if you are filtering and the discarding a lot of messages with very common filter settings. You could use it to filter for an array of property values and even chain comparison operations.

Tags: array[27], dynafile[28], filter[29], if then[30], property[31], template[32]
This is an important new release of the rsyslog v6 devel branch. This release contains bugfixes, as well as new features. The most important one might be the possibility to use the config format for templates.. Please note that part of the feature set is still experimental and might be unstable. ;)

ChangeLog:

http://www.rsyslog.com/changelog-for-6-5-0-v6-devel/[33]

Download:

http://www.rsyslog.com/rsyslog-6-5-0-v6-devel/[34]

As always, feedback is appreciated.

Best regards,
Florian Riedl

Tags: 6.5.0[35], bugfix[36], devel[37], release[38], template[39], templates[40], v6[41]
Version 6.5.0 [devel] 2012-08-28

- imrelp now supports non-cancel thread termination
  (but now requires at least librelp 1.0.1)
- implemented freeCnf() module interface
  This was actually not present in older versions, even though some modules already used it.

The implementation was now done, and not in 6.3/6.4 because the resulting memory leak was ultra-slim and the new interface handling has some potential to seriously break things. Not the kind of thing you want to add in late beta state, if avoidable.

- added –enable-debugless configure option for very high demanding envs
  This actually at compile time disables a lot of debug code, resulting in some speedup (but serious loss of debugging capabilities)
- added new 0mq plugins (via czmq lib)
  Thanks to David Kelly for contributing these modules
- bugfix: omhdfs did no longer compile
- bugfix: SystemLogSocketAnnotate did not work correctly
  Thanks to Miloslav Trmač for the patch
- $SystemLogParseTrusted config file option
  Thanks to Milan Bartos for the patch
- added template config directive
- added new uuid message property
  Thanks to Jérôme Renard for the idea and patches.
  Note: patches were released under ASL 2.0, see
  http://bugzilla.adiscon.com/show_bug.cgi?id=353[42]

Tags: 6.5.0[43], bugfix[44], Changelog[45], devel[46], imrelp[47], rsyslog[48], template[49], templates[50], v6[51]

This little FAQ describe how to bind a template.
First with the new template format "list" and then with the old "legacy" format.

First off all you have to define a template for example for specify output.

Here is an example template in the list format:

```
template(name="FileFormat" type="list") {
property(name="timestamp" dateFormat="rfc3339")
constant(value=" ")
property(name="hostname")
constant(value=" ")
property(name="syslogtag")
constant(value=" ")
property(name="msg" spifno1stsp="on" )
property(name="msg" droplastlf="on" )
constant(value="\n")
}
```

Then you have to bind the template to an action. The Syntax to bind a template is:

> *Action;name-of-template*

Here is an example action with a example-template:

> **.* action(type="omfile" file="/var/log/all-messages.log");Name-of-your-template*

In the configuration it should looks like this:

> *template(name="FileFormat" type="list") {*
> *property(name="timestamp" dateFormat="rfc3339˝)*
> *constant(value=" ")*
> *property(name="hostname")*
> *constant(value=" ")*
> *property(name="syslogtag")*
> *constant(value=" ")*
> *property(name="msg" spifno1stsp="on")*
> *property(name="msg" droplastlf="on")*
> *constant(value="\n")*
> *}*
> *action(type="omfile" file="/var/log/all-msgs.log");FileFormat*
> *"*

Here is an example for the legacy format

Here is an example template in the legacy format:

> *$template ExampleFormat,"%timereported:::date-rfc3339% %HOSTNAME% %msg%"*

Here is an example action with a example-template:

> **.* /var/log/all-messages.log;Your-Template-Name*

In the Configuration it looks like this:

> *"$template ExampleFormat,"%timereported:::date-rfc3339% %HOSTNAME% %msg%"*
> **.* /var/log/all-messages.log;ExampleFormat"*

That's it

Tags: bind a template[52], specify output[53], template[54], use a template[55]
In this scenario, we want to store remote sent messages into a specific local file and forward the received messages to another syslog server. Local messages should still be locally stored.

Things to think about

How should this work out? Basically, we need a syslog listener for TCP and one for UDP, the local logging service and two rulesets, one for the local logging and one for the remote logging.

TCP recpetion is not a build-in capability. You need to load the imtcp plugin in order to enable it. This needs to be done only once in rsyslog.conf. Do it right at the top.

Note that the server port address specified in $InputTCPServerRun must match the port address that the clients send messages to.

Config Statements

```
# Modules
$ModLoad imtcp
$ModLoad imudp
$ModLoad imuxsock
$ModLoad imklog
# Templates
# log every host in its own directory
$template RemoteHost,"/var/syslog/hosts/%HOSTNAME%/%$YEAR%/%$MONTH%/%$DAY%/syslog.log"
### Rulesets
# Local Logging
$RuleSet local
kern.*                                                 /var/log/messages
*.info;mail.none;authpriv.none;cron.none                /var/log/messages
authpriv.*                                              /var/log/secure
mail.*                                                  -/var/log/maillog
cron.*                                                  /var/log/cron
*.emerg                                                 *
uucp,news.crit                                          /var/log/spooler
local7.*                                                /var/log/boot.log
# use the local RuleSet as default if not specified otherwise
$DefaultRuleset local

# Remote Logging
```

```
$RuleSet remote
*.* ?RemoteHost
# Send messages we receive to Gremlin
*.* @@W.X.Y.Z:514
### Listeners
# bind ruleset to tcp listener
$InputTCPServerBindRuleset remote
# and activate it:
$InputTCPServerRun 10514

$InputUDPServerBindRuleset remote
$UDPServerRun 514
```

How it works

The configuration basically works in 4 parts. First, we load all the modules (imtcp, imudp, imuxsock, imklog). Then we specify the templates for creating files. The we create the rulesets which we can use for the different receivers. And last we set the listeners.

The rulesets are somewhat interesting to look at. The ruleset "local" will be set as the default ruleset. That means, that it will be used by any listener if it is not specified otherwise. Further, this ruleset uses the default log paths vor various facilities and severities.

The ruleset "remote" on the other hand takes care of the local logging and forwarding of all log messages that are received either via UDP or TCP. First, all the messages will be stored in a local file. The filename will be generated with the help of the template at the beginning of our configuration (in our example a rather complex folder structure will be used). After logging into the file, all the messages will be forwarded to another syslog server via TCP.

In the last part of the configuration we set the syslog listeners. We first bind the listener to the ruleset "remote", then we give it the directive to run the listener with the port to use. In our case we use 10514 for TCP and 514 for UDP.

Important

There are some tricks in this configuration. Since we are actively using the rulesets, we must specify those rulesets before being able to bind them to a listener. That means, the order in the configuration is somewhat different than usual. Usually we would put the listener commands on top of the configuration right after the modules. Now we need to specify the rulesets first, then set the listeners (including the bind command). This is due to the current configuration design of

rsyslog. To bind a listener to a ruleset, the ruleset object must at least be present before the listener is created. And that is why we need this kind of order for our configuration.

Tags: Guides for rsyslog[56], More complex scenarios[57], rsyslog[58], ruleset[59], syslog[60], TCP[61], template[62], UDP[63]

1. http://www.rsyslog.com/
2. http://www.rsyslog.com/tag/changelog-for-8-3-2-v8-devel/
3. http://www.rsyslog.com/tag/download-v8-devel/
4. http://www.rsyslog.com/tag/8-3-2/
5. http://www.rsyslog.com/tag/bugfix/
6. http://www.rsyslog.com/tag/devel/
7. http://www.rsyslog.com/tag/imrelp/
8. http://www.rsyslog.com/tag/mmexternal/
9. http://www.rsyslog.com/tag/release/
10. http://www.rsyslog.com/tag/rsyslog/
11. http://www.rsyslog.com/tag/template/
12. http://www.rsyslog.com/tag/v8/
13. https://github.com/rsyslog/rsyslog/issues/65
14. https://github.com/rsyslog/rsyslog/issues/70
15. https://github.com/rsyslog/rsyslog/issues/69
16. https://github.com/rsyslog/rsyslog/issues/64
17. https://github.com/rsyslog/rsyslog/pull/63
18. http://www.rsyslog.com/tag/8-3-2/
19. http://www.rsyslog.com/tag/bugfix/
20. http://www.rsyslog.com/tag/changelog/
21. http://www.rsyslog.com/tag/devel/
22. http://www.rsyslog.com/tag/imrelp/
23. http://www.rsyslog.com/tag/mmexternal/
24. http://www.rsyslog.com/tag/rsyslog/
25. http://www.rsyslog.com/tag/template/
26. http://www.rsyslog.com/tag/v8/

27. http://www.rsyslog.com/tag/array/
28. http://www.rsyslog.com/tag/dynafile/
29. http://www.rsyslog.com/tag/filter/
30. http://www.rsyslog.com/tag/if-then/
31. http://www.rsyslog.com/tag/property/
32. http://www.rsyslog.com/tag/template/
33. http://www.rsyslog.com/tag/changelog-for-6-5-0-v6-devel/
34. http://www.rsyslog.com/tag/rsyslog-6-5-0-v6-devel/
35. http://www.rsyslog.com/tag/6-5-0/
36. http://www.rsyslog.com/tag/bugfix/
37. http://www.rsyslog.com/tag/devel/
38. http://www.rsyslog.com/tag/release/
39. http://www.rsyslog.com/tag/template/
40. http://www.rsyslog.com/tag/templates/
41. http://www.rsyslog.com/tag/v6/
42. http://bugzilla.adiscon.com/show_bug.cgi?id=353
43. http://www.rsyslog.com/tag/6-5-0/
44. http://www.rsyslog.com/tag/bugfix/
45. http://www.rsyslog.com/tag/changelog/
46. http://www.rsyslog.com/tag/devel/
47. http://www.rsyslog.com/tag/imrelp/
48. http://www.rsyslog.com/tag/rsyslog/
49. http://www.rsyslog.com/tag/template/
50. http://www.rsyslog.com/tag/templates/
51. http://www.rsyslog.com/tag/v6/
52. http://www.rsyslog.com/tag/bind-a-template/
53. http://www.rsyslog.com/tag/specify-output/
54. http://www.rsyslog.com/tag/template/
55. http://www.rsyslog.com/tag/use-a-template/
56. http://www.rsyslog.com/tag/guides-for-rsyslog/
57. http://www.rsyslog.com/tag/more-complex-scenarios/

58. http://www.rsyslog.com/tag/rsyslog/
59. http://www.rsyslog.com/tag/ruleset/
60. http://www.rsyslog.com/tag/syslog/
61. http://www.rsyslog.com/tag/tcp/
62. http://www.rsyslog.com/tag/template/
63. http://www.rsyslog.com/tag/udp/