

NAME

msoak – consume messages from arbitrary MQTT brokers and subscriptions

SYNOPSIS

msoak *configuration*

DESCRIPTION

msoak is a utility with which to simultaneously subscribe to an arbitrary number of topics on any number of MQTT brokers and optionally modify or normalize received payloads before printing them out. This utility was created to back up to a central location messages received by a hand full of brokers; instead of launching (and having to monitor success of) a large number of *mosquitto_sub*(1) programs, *msoak* took on the job.

msoak uses asynchronous connects to the MQTT brokers so that it can handle situations in which a broker may temporarily be unavailable even when *msoak* initially starts, and *msoak* embeds a Lua interpreter for which you can create scripts which act on messages received to, for example, trigger actions.

msoak can be used with, say, *tinylog*(8) to create compressed archives of data.

```
msoak my.config | tinylog -k 30 -s 1077936128 -z archive
```

OPTIONS

None.

CONFIGURATION

msoak reads the *configuration* file using *libconfig*(3) utility functions. This is a short example:

```
# this is a comment

luascript = "example.lua"

servers = (
    { id = "loc"
      host = "localhost"
      fmt = "greet"
      topics = [
          "tests/sponge/#"
      ]
    },
    { id = "tmo"
      host = "test.mosquitto.org"
      qos = 0
      showid = true
      showtopic = true
      topics = [
          "home/special",
          "nothome/dont"
      ]
    }
);
```

The configuration consists of **global** settings and a list or array of **servers**, each with a number of settings.

global variables

luascript the path to the Lua script file to use. If left unset, no Lua processing is performed.

verbose	whether or not to be verbose (default true)
server array	
id	identifier for the connection, also shown on output. If <i>id</i> is unset we generate a numeric <i>id</i> .
host	host name or address of the MQTT broker. Defaults to localhost.
port	the TCP port number for the MQTT broker. Defaults to 1883.
clientid	MQTT clientId to use. The client identifier defaults to the string <i>msoak-hostname:base-name(configfilename)</i> so that the program can run on multiple hosts with the same configuration.
qos	MQTT QoS (default 1)
user	username for the MQTT connection
pass	password for the MQTT connection; see <i>passenv</i> to omit from having to specify clear-text passwords in the <i>configuration</i> file.
passenv	name of environment variable (including \$) from which to obtain <i>pass</i> .
showretained	consume retained messages; default true
cacert	path to PEM-encoded CA certificate chain for TLS connections to MQTT broker
showid	true or false, default true; whether to print <i>id</i> on output.
showtopic	true or false, default true; whether to print topic name on output.
topics	array of strings with topic branch names for <i>msoak</i> to subscribe to.
fmt	optional name of Lua formatting function (see below).

Formatting

By default, the received payload is printed to standard output, optionally prefixed by the message topic (if *showtopic* is true), and it is preceeded by the connection identifier if *showid* is true.

If *fmt* is set, it contains a string with the name of a Lua function from the Lua script specified in the *luascript* global. this function is used to format the payload *msoak* receives; the return value of the function replaces the original payload and is printed out.

If *msoak* can decode the payload into JSON (i.e. the message begins with a brace (`{`) and the JSON can be decoded), it will invoke the Lua function to obtain output.

```
function init()
    if msoak.verbose then
        msoak.log(msoak.luascript .. " starting up")
    end
end

function greet(topic, _type, d)
    s = string.format("Hello %s -> now=%s", d.name,
        msoak.strftime("%TZ"))
    return s
end

function exit()
    msoak.log("Hasta la vista, baby!")
end
```

The optional *init()* and *exit()* functions are invoked when *msoak* begins and ends respectively. The *greet* function is invoked for each message for which a server in the *configuration* file contains **fmt** = "**greet**".

Lua functions

There are a few variables and functions *msoak* implements which are available to the Lua scripts you use.

version	returns the <i>msoak</i> version number as a string
luascript	returns the file name of the <i>luascript</i> global variable
verbose	is a boolean which indicates whether <i>msoak</i> is running in verbose mode
msoak_log()	accepts a string which is printed to <i>stderr</i> prefixed by "MSOAKLOG:".
msoak_strftime()	expects a <i>format</i> string and integer <i>seconds</i> and implements <i>strftime(3)</i> for Lua with the specified format and seconds and returns the string result to Lua. As a special case, if <i>seconds</i> is less than one it uses current time.

JSON

When configured to use a Lua script file, *msoak* attempts to decode incoming JSON payloads and will pass the decoded JSON elements to the configured *fnt* function as a table with these additional elements in it

_conn_id	the original connection <i>id</i>
_conn_host	the hostname of the connection
_conn_port	the port number of the connection
_conn_topic	the MQTT topic on which the original payload message was received

Note that if no *luascript* was specified and the payload contains JSON it will be dumped as is to *stdout*.

ENVIRONMENT

Any number of environment variables may be used by *msoak* if specified in *passenv* settings with the configuration.

BUGS

What's with the strange name? Just as I started working on this program I learned about *sponge(1)* and loved the name. The rest is history.

Note that there are different versions of *libconfig(3)* floating around which may have effects on the syntax permitted in *msoak*'s *configuration* file .

AUTHOR

Jan-Piet Mens, <https://github.com/jpmens/msoak>

SEE ALSO

mosquitto_sub(1), mqttwarn, sponge(1), strftime(1), tinylog(8)