

Summary

This document explains and details how to configure Nagios, using Groundwork Monarch, to forward specific error conditions as syslog messages to Microsoft MOM servers

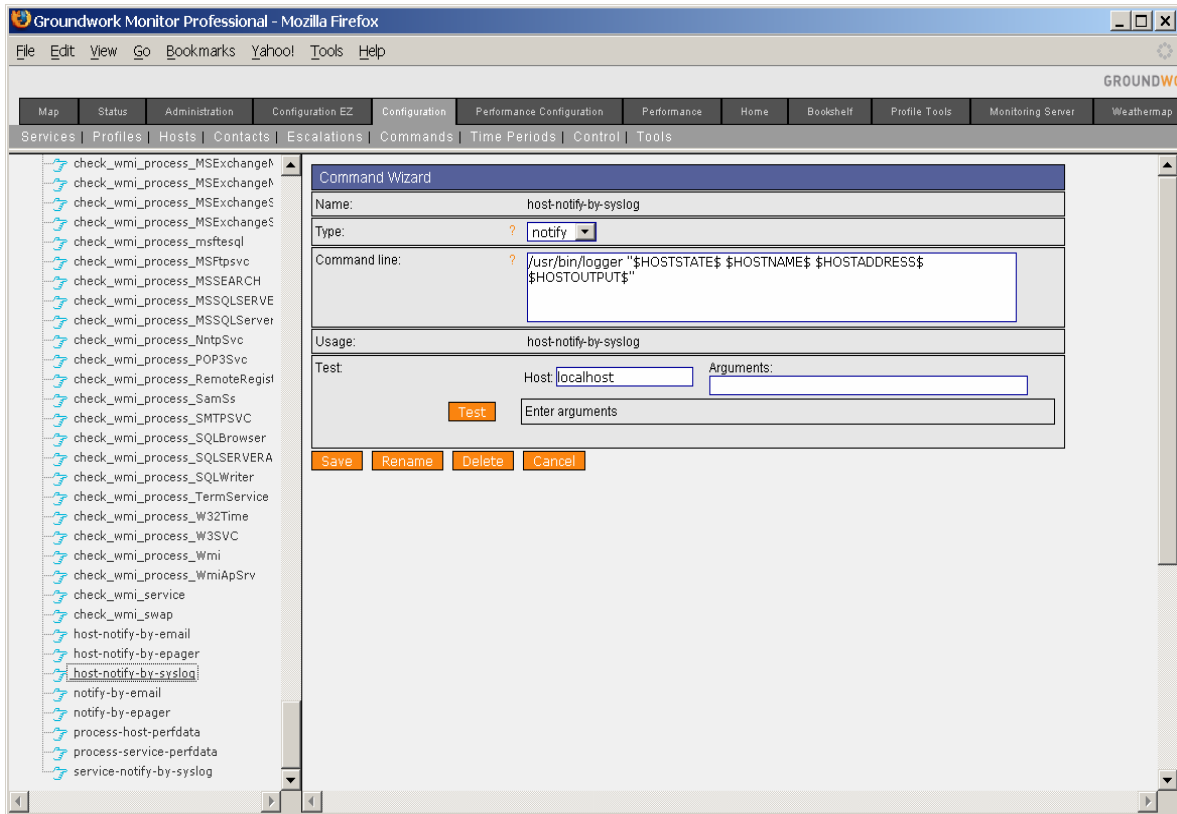
Installation Dependencies

- Groundwork Monitor
 - o Syslog-ng configuration expertise is required. Contact Groundwork support for assistance modifying local syslog-ng configuration to forward syslog messages to MOM server
 - See (http://www.groundworkopensource.com/wiki/index.php/Syslog-ng.conf_Example) for background information
 - See (http://www.cisco.com/univercd/cc/td/doc/product/rtrmgmt/oer/1_1/mce/user/syslog.htm) for assistance with Linux Syslog configuration.

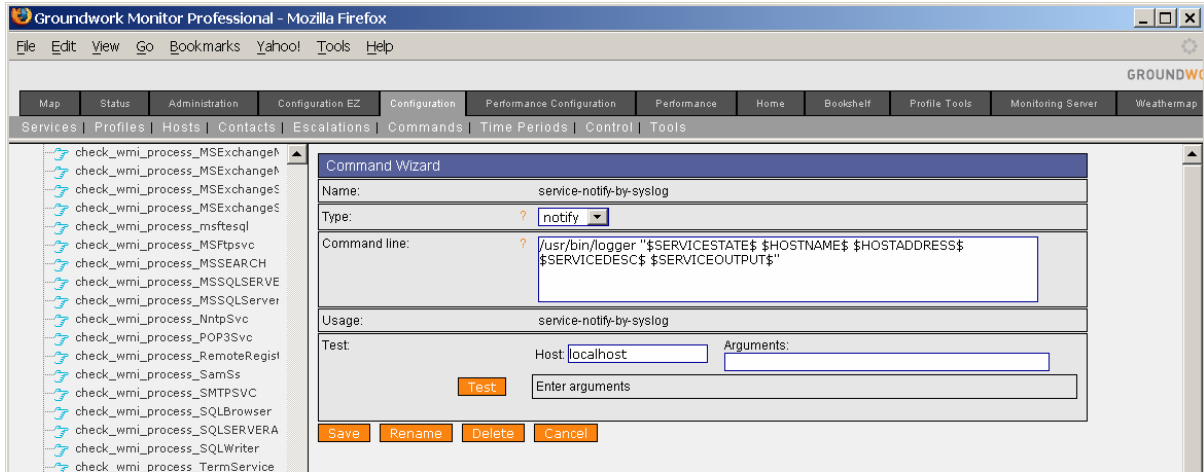
- MOM
 - o See (http://msdn.microsoft.com/library/default.asp?url=/library/en-us/mom/sdk/HTM/P/MOM_P_Interop.asp) for specific dependencies.
 - Groundwork setup assumes the Windows server can receive standard syslog messages

Nagios configuration via Monarch

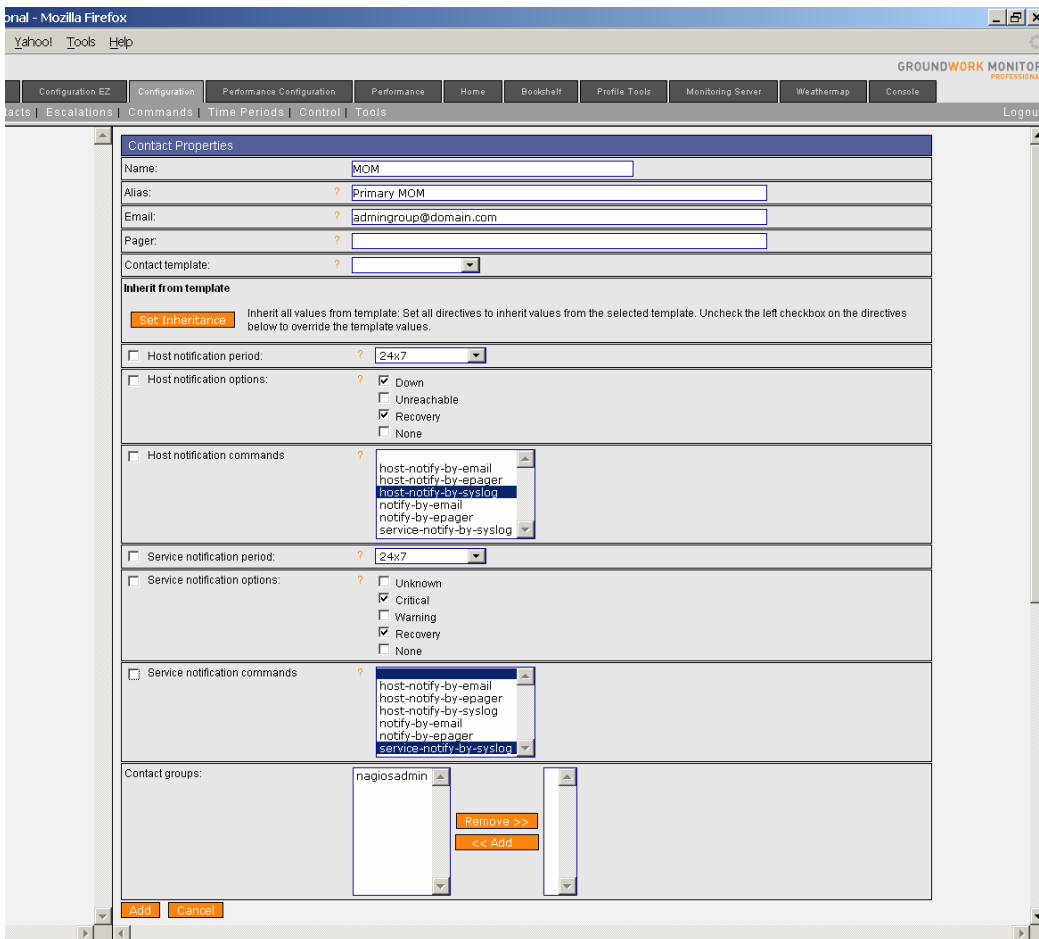
- Copy host-notify-by-mail from Configuration -> Commands
Change values to those below



Repeat same process, but rename as follows:



- Create MOM as a contact in Groundwork Monitor



Setup the options you wish, but make sure that you uncheck the left box so you can override the template and choose the new options for syslog notification instead of email. Commit your changes from Configuration -> Control -> Commit

MOM will now receive every alert that is configured to be sent to the nagiosadmin group.

Defining Event Handling Conditions

Now tune your alerts to just host up/down or service critical/ok changes if you wish by modifying the templates used to construct Services. The below example shows how a Unix check from Nagios can send MOM syslog events, once MOM is configured as a contact inside of Groundwork Monitor

The screenshot shows the configuration page for a service named 'ssh_load'. The left sidebar lists various services, with 'ssh_load' selected. The main configuration area includes the following settings:

- Check freshness: ?
- Freshness threshold: ? []
- Notifications enabled: ?
- Notification interval: ? [120]
- Notification period: ? [24x7]
- Notification options: ?
 - Unknown
 - Critical
 - Warning
 - Recovery
 - None
- Event handler enabled: ?
- Event handler: ? []
- Flap detection enabled: ?
- Low flap threshold: ? []
- High flap threshold: ? []
- Process perf data:
- Retain status information: ?
- Retain nonstatus information: ?
- Contact Groups: [nagiosadmin]
 - Remove >>
 - << Add
- Extended info template: [unix_load_graph]
- Escalation tree: ? []

Buttons at the bottom: Save, Delete, Rename, Close.

Choose status for alert condition for either host or service. Review Nagios documentation (http://nagios.sourceforge.net/docs/2_0/macros.html#hoststate) to explore additional data available as plugin output that is easily passed as a macro, which is called inside of the Monarch configuration database entry for MOM Event Handling via Nagios.

Configuring Groundwork Sever Syslog Options

If your Groundwork host is running standard syslog, here is a suggested configuration file

```
# cat mom.syslog-ng.conf

# $Id: client-syslog-ng.conf,v 1.4 2005/10/23 18:36:10 jmates Exp $
#
# syslog-ng client configuration: some local logs, in addition to TCP
# logging to central loghost. Listens only on localhost interface;
# requires "logs" user and group on system.
#
# Local logs are stored under /var/log/archive in a syslog-ng specific
# format that includes facility, priority, and a timestamp that includes
# the year.

options {
  log_fifo_size(4096);

  group(logs);
  dir_group(logs);

  create_dirs(yes);
  dir_perm(0750);
  perm(0640);
  use_time_recvd(no);

  use_fqdn(no);
  chain_hostnames(no);
  keep_hostname(yes);

  stats(3600);
};

source local {
  unix-stream("/dev/log" max_connections(150));
  udp(ip(127.0.0.1) port(514));
  internal();
};

# Enable either UDP or TCP option based on MOM configuration
# all logs to MOM via UDP
filter notdebug { level(info...emerg); };
destination loghost { udp("172.28.108.8" port(514)); };
```

```
log { source(local); filter(notdebug); destination(loghost); };
```

```
# all logs to MOM via TCP
```

```
#filter notdebug { level(info...emerg); };
```

```
#destination loghost { udp("172.28.108.8" port(514)); };
```

```
#log { source(local); filter(notdebug); destination(loghost); };
```

```
# Created 14SEP06
```

Update the file with MOM's IP address and choose protocol and port.

Backup your existing syslog.conf and then replace with MOM config

```
#cp /etc/syslog.conf /etc/bak.syslog.conf
```

```
#cp /etc/mom.syslog.conf /etc/syslog.conf
```

```
#service syslog restart
```

Test MOM is receiving messages with

```
#!/usr/bin/logger hello mom
```

This message should appear in the MOM console from

Configuring Groundwork Sever Syslog-NG Options

If your Groundwork host is running syslog-ng, here is a suggested configuration file

```
#cat mom.syslog-ng.conf
# The syslog-ng.conf configuration file uses blocks (options, source, filter,
destination, and log)
# that together specify options, and how log entries are gathered, matched, and
routed.
#

# Options

options {
#group(logs);
#dir_group(logs);

#perm(0640);
#dir_perm(0750);
#create_dirs(yes);

#log_fifo_size(4096);

#use_fqdn(yes);
keep_hostname(yes);
chain_hostnames(no);

#stats(3600);

#bad_hostname("gconfd");
};

# Sources
source src { unix-stream("/dev/log"); internal(); };
source kernsrc { file("/proc/kmsg"); };
source s_udp { udp(); };

# Set up host specific log files
# Local file write is disabled below and forced to MOM
#destination messages { file("/var/log/messages"); };
#destination kern { file("/var/log/kern.log"); };

# Choose option below to use when forwarding events to MOM
#destination loghost {
#tcp("loghost.example.org" port (5000));
#};
```



```
destination loghost {  
  udp("172.28.108.8" port (514));  
};
```

Destinations

The destination block is used to send logs somewhere, whether to a file, remote host, or program.

Use these files to run Nagios checks for error and forward to MOM when appropriate

```
destination host_splitter { file(/usr/local/groundwork/var/log/syslog-ng/$HOST.log); };
```

Logs

log blocks to join source to destination with optional filter specifications.

```
log { source(s_udp); destination(host_splitter); };
```

```
log { source(kernsrc); destination(kern); };
```

```
log { source(src); destination(messages); };
```