

## Nagios Monitoring of Switches and Routers

This example shows how the Nagios monitoring tool can be used to monitor the message count of a queue and the free memory of a Router or Switch using the OID from an SNMPwalk.

```
nagios.cfg In Nagios' main configuration file we add a new object configuration file where we define the entries to
monitor the SwiftMQ Router: cfg_file=/usr/local/nagios/etc/objects/swiftmq.cfg First, we define the host we want to
monitor (the machine where the SwiftMQ Router runs): define host{ use generic-host ; Name of
; This host definition will inherit all variables that are defined ; in (or inherited by) the linux
definition. max_check_attempts 10 host_name msgserver1 alias msgserver1
define the host group "SwiftMQ-Servers" where this machine belongs to: define hostgroup{ hostgroup_name Swift
Servers ; The name of the hostgroup alias SwiftMQ Servers ; Long name of the group members m
separated list of hosts that belong to this group } A command "check_snmpv2c" to check with SNMPv2: define
command{ command_name check_snmpv2c command_line $USER1$/check_snmp -H $HOSTADDRESS$ -
$ARG1$ } And the service to monitor testqueue's messagecount. Note we use port 10061, the OID we determined with
an SNMP walk on the machine (google it if need be), and set a warning level at 8000 and a critical threshold at 10000
messages: define service{ use generic-service ; Inherit values from a template host_name msgserver1
service_description testqueue message count check_command check_snmpv2c!-p 10061 -C public -o
1.3.6.1.4.1.16683.1.2.11.1.12.1.1.10.9.116.101.115.116.113.117.101.117.101 -c 10000 -w 8000 } Finally the service to
monitor the free memory of the router: define service{ use generic-service ; Inherit values from a template host
msgserver1 service_description Free Memory SwiftMQ router1 check_command check_snmpv2c!-p 10061 -C public
o 1.3.6.1.4.1.16683.1.2.1.1.10.1.1.2.7.114.111.117.116.101.114.49!5000!1000 } Test Now start Nagios, open the
Nagios web console and send messages to "testqueue" (simply use the example "P2PSender" in
samples/router_network
```