

# GDMA Troubleshooting

## Contents

This page reference troubleshooting tips for GDMA.

## 1.0 Common Issues

### 1.1 Target Side

- Are the appropriate **ports** accessible?
  - If auto-registration is completing but monitoring data from GDMA clients is obviously not being processed by the server, perhaps the problem is that the server does not have the Bronx listener port (by default, port 5667) open to the clients. See the `listener_port` value in the `/usr/local/groundwork/config/bronx.cfg` file on the server, and the `Spooler_NSCA_Port` value in the `/usr/local/groundwork/gdma/config/gdma_auto.conf` file on the GDMA client. These values must match, and you must not have a firewall (or packet filtering such as iptables on the server) blocking this port.
  - The mechanism for allowing access to a given server port will depend on the particular Linux distribution you are running on the server. On RHEL 6.3, for instance, you would need a line such as the following in the `/etc/sysconfig/iptables` file, and then have iptables restarted to pick up this configuration:

```
-A INPUT -m state --state NEW -m tcp -p tcp --dport 5667 -j ACCEPT
```

- **Time Sync** - Are all machine clocks synchronized? Many installations do not properly synchronize time with NTP. This is particularly an issue with Virtual environments, where time slicing may introduce latency and drift to otherwise stable time signatures. For this reason, GDMA has been made tolerant of time drift. In the setup instructions, the file `/usr/local/groundwork/config/bronx.cfg` was modified to allow up to 900 seconds latency and 90 seconds imminence, or future timestamps, on incoming messages. If the time synchronization is sufficiently off to be outside this 16.5 minute envelope, the GroundWork server will reject the incoming message. To correct this situation, you must either:



The GroundWork Monitor version of `send_nsc` supplied with GDMA allows for the sending of old time stamps, which are the time stamps associated with when the check was run, not when the spooler has sent the data, which is potentially much later. This allows the system to accurately process performance data for checks that have been spooled, potentially for long periods.

1. Synchronize the time on all monitored systems and the GroundWork server, or
  2. Widen the envelope by changing the parameters in `bronx.cfg`. It is highly recommended that the time be synchronized. You will need to restart Nagios after you make a change, `service groundwork restart nagios`.
- **Encryption Mismatch** - A common security requirement that GroundWork Monitor supports is the encryption of data sent over the NSCA channel. By default, however, GDMA is installed with encryption turned off, which is also the default on the GroundWork server. Thus it is possible to have an encryption mismatch, for example, if you enable encryption on the GroundWork server (in `bronx.cfg`), and fail to enable it on the GDMA in `send_nsc.cfg`. Always make sure these files are consistent between GDMA (`send_nsc.cfg`) and GroundWork (`bronx.cfg`).
    - One way to manage this situation is to set up GDMA to use an alternate `send_nsc.cfg` file. This file can be downloaded to the GDMA as if it were a plugin, and specified for use via the host external using the GDMA parameter called `Spooler_NSCA_Config`.
      - For example, if you create a file called:

```
send_nsc_gdma_encrypted.cfg
```

and you set it up for download as a plugin for Linux, it will be placed on the Linux GDMA in:

```
/usr/local/groundwork/gdma/tmp/send_nsc_gdma_encrypted.cfg
```

that means you could set up the GDMA to start using this file at the same time you switch the GroundWork server to use encryption, just by changing:

```
Spooler_NSCA_Config="/usr/local/groundwork/gdma/tmp/send_nsca_gdma_encrypted.cfg"
```

- Are **externals** enabled? - The configuration for GDMA monitored hosts must have Externals enabled to allow access to all the externals-related screens and options. Depending on the version of GroundWork Monitor you are using this directive may already be enabled. To verify/enable externals go to *Configuration > Control > Setup > Enable externals*.
- Was the setup of **Auto Registration** for the GDMA target server done correctly?
  - Verify the *Auto-Registration* host group exists using *Configuration > Hosts > Host groups > Modify*.
  - Verify the *auto-registration* monarch group exists using *Configuration > Groups > Groups > auto-registration > Detail*.
  - Is the build folder entry correct?
    - The Build folder is the location where externals files will be placed and where GDMA Clients will be looking to pick them up, *Configuration > Groups > Groups > auto-registration > Detail > Build folder*. The build folder is usually `/usr/local/groundwork/nagios/etc/<hostname>` where you have replaced `<hostname>` with the actual standby or child server hostname used to name the group. In general, this may be any directory writeable by the 'nagios' user where you wish to store the configuration files. An incorrect entry or a directory with incorrect permissions will cause the build instance to fail. For a Monarch group used simply to reference hosts for generating externals, you must specify `/usr/local/groundwork/apache2/htdocs/<GDMAConfigDir>` instead, where you have replaced `<GDMAConfigDir>` with the value of the GDMAConfigDir option in the GDMA client `gdma_auto.conf` config file (usually just 'gdma').
  - Example target server listing of config files of GDMA hosts

```
[root@ip-10-0-10-109 config]# cd /usr/local/groundwork/apache2/htdocs/gdma
[root@ip-10-0-10-109 gdma]# ls
gwmon_ip-10-0-10-209.localdomain.cfg  gwmon_ip-10-0-10-216.cfg  gwmon_ip-10-0-10-219.cfg
gwmon_ip-10-0-10-79.cfg
```

- Verify or add the *Auto-Registration* host group as part of the *auto-registration* monarch group. Go to *Configuration > Groups > Groups > auto-registration > Detail > select the Hosts tab*. Scroll down to *Add Hostgroup(s)*, check and add *Auto-Registration*.
- **GDMA Profiles**
  - Was the GDMA target server configured with the GDMA profiles? *Configuration > Profiles > Profile importer > Import > GDMA*.
  - Were the associated GDMA profiles host templates assigned appropriate Contact Groups? *Configuration > Hosts > Host templates > Modify*, (*gdma-aix-host*, *gdma-linux-host*, *gdma-solaris-host*, *gdma-windows-host*) verify/add contact group.
- Did you perform a **Commit** and **Build externals** operations after installing the GDMA on all clients?
  - Just like the addition of any other host, we need to run a *Configuration > Control > Commit* operation to allow the monitoring engine to recognize the host and see it in Status. Since it is a GDMA host you will need to be in the habit of running a *Build externals* operation (from that same menu list) so that GDMA configuration changes can be picked up by GDMA hosts.
  - If you see that the nagios log reports the new configuration with an alert, perhaps because you haven't committed the new definition and Nagios is not recognizing it as a valid host, or maybe profile services cannot be found.

## 1.2 Client Side

- Is there a **pre-existing host**?\*\* The `/usr/local/groundwork/gdma/config/gdma_override.conf` file is new as of the GDMA 2.3.0 release. It is dynamically created as the place where the poller will store the hostname returned by the server (e.g. `gwmon_ip-10-0-10-209.localdomain.cfg`) after a successful auto-registration request. The poller uses a `Forced_Hostname` option to save this centrally-agreed-upon hostname. This gets picked up by the spooler as well, so both poller and spooler know the registered name of the host. Any use of the `Forced_Hostname` option in the `gdma_auto.conf` file will be overridden by the value in the `gdma_override.conf` file, if the latter file is present.

Example client config files:

```
[root@ip-10-0-10-209 ~]# cd /usr/local/groundwork/gdma/config
[root@ip-10-0-10-209 config]# ls
gdma_auto.conf      gwmon_ip-10-0-10-209.localdomain.cfg  send_nsca.cfg
gdma_override.conf  multihost_gdma_auto.conf
[root@ip-10-0-10-209 config]# cat gdma_override.conf
# DO NOT EDIT: The contents of this file are subject to being overwritten,
# so you should not consider this data to be permanently persistent.
Forced_Hostname = "ip-10-0-10-209.localdomain"
```

- If an Auto-Registration host does not show up there could be a possible pre-existing host. There is a file on the GDMA client in the `/usr/local/groundwork/gdma/config` directory named `gwmon_override.conf` which will be written under conditions that the server side recognizes. In some cases it is because the IP address and name of the host pre-existed on the GroundWork side, and the auto registration process concluded that the machine name is what did exist and goes with that name. The agent side accepts it and writes the name into that file, and it never goes away. To try and fix, delete the file and restart the GDMA service and the correct name should be used. After deleting the configuration files take a look at Monarch and see if you can see the correct host name, also perform a commit to populate the Status view.

- Viewing the **gdma\_auto.conf** configuration file.
  - The new values requested by the GDMA client installer are stored on the client in the `gdma_auto.conf` file. These settings can be adjusted after an installation, if needed. Like anything else in the `gdma_auto.conf` file, modifying values will cause the client to recognize a configuration change and attempt to reload its own configuration, in up to three different files; `gdma_auto.conf`, `gdma_override.conf`, and the `gwmon_hostname.conf`. You can check to see if the GDMA installed properly on the client(s), if the correct registration username and password were used for Auto Registration, and if the correct name of the GDMA target server (GroundWork Server) was entered?
    - **For Windows:** C:\Program Files (x86)\groundwork\gdma\config\gdma\_auto.conf
    - **For Linux:** /usr/local/groundwork/gdma/config/gdma\_auto.conf

Example `gdma_auto.conf` on Linux server

```

[root@ip-10-0-10-209 ~]# cd /usr/local/groundwork/gdma/config
[root@ip-10-0-10-209 config]# ls
gdma_auto.conf      gwmon_ip-10-0-10-209.localdomain.cfg  send_nsca.cfg
gdma_override.conf  multihost_gdma_auto.conf
[root@ip-10-0-10-209 config]# cat gdma_auto.conf
#####
# gdma_auto.conf
#
# Copyright 2004-2014 GroundWork Open Source, Inc.
# [http://www.groundworkopensource.com]
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
#
#####
Auto_Register_Attempts = "fibonacci"
Auto_Register_Host_Profile = "gdma-linux-host"
Auto_Register_Max_Interval = "86400"
Auto_Register_Pass = "gdma"
Auto_Register_Service_Profile = ""
Auto_Register_User = "gdma"
ConfigFile_Pull_Cycle = "1"
ConfigPull_Timeout = "10"
Enable_Auto = "on"
Enable_Local_Logging = "off"
GDMAConfigDir = "gdma"
GDMA_Auto_Host = "gdma-autohost"
GDMA_Auto_Service = "gdma_auto"
GDMA_Multihost = "off"
Logdir = "/usr/local/groundwork/gdma/log/"
Max_Concurrent_Hosts = "1"
Max_Server_Redirects = "5"
PluginPull_Timeout = "40"
Poller_Plugin_Directory = "/usr/local/groundwork/nagios/libexec"
Poller_Plugin_Timeout = "5"
Poller_Proc_Interval = "600"
Poller_Pull_Failure_Interval = "86400"
Poller_Service = "gdma_poller"
Poller_Status = "On"
Spooler_Batch_Size = "20"
Spooler_Max_Retries = "10"
Spooler_NSCA_Config = "/usr/local/groundwork/gdma/config/send_nsca.cfg"
Spooler_NSCA_Port = "5667"
Spooler_NSCA_Program = "/usr/local/groundwork/common/bin/send_nsca"
Spooler_NSCA_Timeout = "5"
Spooler_Proc_Interval = "30"
Spooler_Retention_Time = "900"
Spooler_Service = "gdma_spooler"
Spooler_Status = "Updates"
Target_Server = "http://10.0.10.109"

# The Use_Long_Hostname option actually has 3 supported states: "on", "off",
# and not-defined (i.e., commented out), which invokes a fallback behavior.
# Use_Long_Hostname = "on"

# The Use_Lowercase_Hostname option used to be internally hardcoded as "on",
# but now is defaulted "off". If you still want the legacy behavior, you can
# override that default value here.
# Use_Lowercase_Hostname = "on"
[root@ip-10-0-10-209 config]#

```

## 2.0 Generating Logs

Logs can be viewed on both the server side and client side.

- **Server Side**

- **Nagios Log** - The nagios log is located in the `/usr/local/groundwork/nagios/var` directory. After a commit operation, the auto registration process begins and a nagios log message will announce the new GDMA. As soon as the auto-registration cron job runs, the message is matched to the profile, the host definition is created with that profile, and the Monarch group gets the host added.

Example GDMA server nagios.log:

```
[root@ip-10-0-10-109 ~]# cd /usr/local/groundwork/nagios/var
[root@ip-10-0-10-109 var]# ls
archives          nagios.lock      objects.cache    status.log
checkresults     nagios.log       rw
COMMIT_LOCKFILE  nagios_seek.tmp  service-perfdata.dat
log              nagiosstatus.sav spool
[root@ip-10-0-10-109 var]# tail nagios.log
[1491584348] PASSIVE SERVICE CHECK: ip-10-0-10-209.localdomain;gdma_poller;0;OK Poller
processed 6 checks in 0.161 secs. Using 0.03% of 600 sec Polling Interval.
[1491584378] PASSIVE SERVICE CHECK: ip-10-0-10-209.localdomain;gdma_spooler;0;Spooler
transmitted 7 results in 0.008 secs
[1491584868] PASSIVE SERVICE CHECK: ip-10-0-10-219;gdma_21_wmi_disk_C;1;Warning - C:
'Percentage Space Used' = 88:
[1491584868] PASSIVE SERVICE CHECK: ip-10-0-10-219;gdma_21_wmi_uptime;1;WARNING: Uptime
(s) = 1520. Host name = IP-10-0-10-219, domain = WORKGROUP, OS Name = Microsoftr
Windows Serverr 2008 Datacenter Service Pack 2, manufacturer = Xen, model = HVM domU,
processors = 1, Last boot time (local clock) = 2017-04-07 09:41:58
[1491584868] PASSIVE SERVICE CHECK: ip-10-0-10-219;gdma_21_wmi_disktransfers;0;OK -
_Total 'DiskTransfersPersec' = 0:
[1491584868] PASSIVE SERVICE CHECK: ip-10-0-10-219;gdma_21_wmi_memory_pages;0;OK -
PagesPerSec 'PagesPerSec' = 0:
[1491584868] PASSIVE SERVICE CHECK: ip-10-0-10-219;gdma_21_wmi_mem;0;OK - _Total
'Percentage Space Used' = 22:
[1491584868] PASSIVE SERVICE CHECK: ip-10-0-10-219;gdma_21_wmi_disk_all;1;Warning - C:
'Percentage Space Used' = 88: OK - Z: 'Percentage Space Used' = 1:
[1491584868] PASSIVE SERVICE CHECK: ip-10-0-10-219;gdma_21_wmi_cpu;0;OK - _Total 'Load
Percentage' = 0:
[1491584868] PASSIVE SERVICE CHECK: ip-10-0-10-219;gdma_poller;0;OK Poller processed 7
checks in 4.150 secs. Using 0.69% of 600 sec Polling Interval.
```

- **Register Agent Log** - If you think that auto registration is not doing what it ought to, you can look in the log file `register_agent.log`.

Example Register Agent Log:

```
cat /usr/local/groundwork/foundation/container/logs/register_agent.log
```

```
(23743) === Starting up (process 23743). ===
(23743) NOTICE: Received auto-registration request with these host attributes:
agent_type = GDMA
host_name = ip-10-0-10-209.localdomain
host_address = 10.0.10.209
host_mac = 06:2A:81:9A:C3:86
host_os = linux
host_profile = gdma-linux-host
service_profile =
(23743) STATS: Host "ip-10-0-10-209.localdomain", address "10.0.10.209" has been added to
Monarch.
(23743) STATS: Hostgroup "Auto-Registration" has been assigned to host
"ip-10-0-10-209.localdomain".
(23743) STATS: Externals have been created for host "ip-10-0-10-209.localdomain".
(16624) === Starting up (process 16624). ===
(16624) NOTICE: Received auto-registration request with these host attributes:
agent_type = GDMA
host_name = ip-10-0-10-219
host_address = 10.0.10.219
host_mac = 06:D 3:08:DE:DC:46
host_os = windows
host_profile = gdma-windows-host
service_profile =
(16624) WARNING: Found existing host "ip-10-0-10-219" (10.0.10.219) with undefined OS,
(16624) while registering ip-10-0-10-219 (10.0.10.219) with OS "windows".
(16624) Will assume a match, without overwriting the existing value.
(16624) STATS: Host profile "gdma-windows-host" has been applied to host
"ip-10-0-10-219".
(16624) STATS: Hostgroup "Auto-Registration" has been assigned to host "ip-10-0-10-219".
(16624) STATS: Externals have been created for host "ip-10-0-10-219".
```

- **Client Side**

- **Enable Logging** - Auto-registration error messages listed in the server log are generally also returned to the GDMA client, and will be recorded in the client's poller log file if logging is enabled on the client to output log data to the file specified in the *Logdir* parameter, which is the log subdirectory by default. If the client is having trouble contacting the server in the first place, that may not be evident without enabling logging on the client and looking in the poller log file. If you do enable logging on the client, it is highly recommended that you disable it once you have diagnosed and fixed the problem, because the current version of GDMA never rotates its log files.

- Enable\_Local\_Logging = "on"

Example `gdma_auto.conf` logging parameter:

```

[root@ip-10-0-10-209 log]# cd /usr/local/groundwork/gdma/config
[root@ip-10-0-10-209 config]# ls
gdma_auto.conf      gwmon_ip-10-0-10-209.localdomain.cfg  send_nsca.cfg
gdma_override.conf multihost_gdma_auto.conf
[root@ip-10-0-10-209 config]# vi gdma_auto.conf
#####
#   gdma_auto.conf
#
#   Copyright 2004-2014 GroundWork Open Source, Inc.
#   http://www.groundworkopensource.com
#
#   Unless required by applicable law or agreed to in writing, software
#   distributed under the License is distributed on an "AS IS" BASIS,
#   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
#   See the License for the specific language governing permissions and
#   limitations under the License.
#
#####
Auto_Register_Attempts = "fibonacci"
Auto_Register_Host_Profile = "gdma-linux-host"
Auto_Register_Max_Interval = "86400"
Auto_Register_Pass = "gdma"
Auto_Register_Service_Profile = ""
Auto_Register_User = "gdma"
ConfigFile_Pull_Cycle = "1"
ConfigPull_Timeout = "10"
Enable_Auto = "on"
Enable_Local_Logging = "off"
GDMAConfigDir = "gdma"
GDMA_Auto_Host = "gdma-autohost"
GDMA_Auto_Service = "gdma_auto"
GDMA_Multihost = "off"
Logdir = "/usr/local/groundwork/gdma/log/"
Max_Concurrent_Hosts = "1"
Max_Server_Redirects = "5"
PluginPull_Timeout = "40"
Poller_Plugin_Directory = "/usr/local/groundwork/nagios/libexec"
Poller_Plugin_Timeout = "5"
Poller_Proc_Interval = "600"
Poller_Pull_Failure_Interval = "86400"
Poller_Service = "gdma_poller"
Poller_Status = "On"
Spooler_Batch_Size = "20"
Spooler_Max_Retries = "10"
Spooler_NSCA_Config = "/usr/local/groundwork/gdma/config/send_nsca.cfg"
Spooler_NSCA_Port = "5667"
Spooler_NSCA_Program = "/usr/local/groundwork/common/bin/send_nsca"
Spooler_NSCA_Timeout = "5"
Spooler_Proc_Interval = "30"
Spooler_Retention_Time = "900"
Spooler_Service = "gdma_spooler"
Spooler_Status = "Updates"
Target_Server = "http://10.0.10.109"

# The Use_Long_Hostname option actually has 3 supported states: "on", "off",
# and not-defined (i.e., commented out), which invokes a fallback behavior.
# Use_Long_Hostname = "on"

# The Use_Lowercase_Hostname option used to be internally hardcoded as "on",
# but now is defaulted "off".  If you still want the legacy behavior, you can
# override that default value here.
# Use_Lowercase_Hostname = "on"
[root@ip-10-0-10-209 config]#

```

Example client side log directory:

```
[root@ip-10-0-10-209 log]# cd /usr/local/groundwork/gdma/log
[root@ip-10-0-10-209 log]# ls
gwmon_ip-10-0-10-209.localdomain_poller.log
gwmon_ip-10-0-10-209.localdomain_run_checks.log
gwmon_ip-10-0-10-209.localdomain_spool_processor.log
NOTEEMPTY
```

- **Running by Hand** - You can also generate useful output (the same as what gets placed in the log) by running the GDMA Poller and/or Spooler by hand.

The spooler merely takes output from the checks that have been run and sends it to the target server or servers. Of interest here is if it succeeded in contacting the Target, which will be indicated as Processing live targets. Dead targets are target servers that can't be reached for some reason, and indicate a down primary or standby system, or network issues in contacting the target. The spooler will keep trying until it reaches a dead target, or times out according to the `Spooler_Retention_Time` and `Spooler_Max_Retries` parameters.

For UNIX, the command is similar, but you must run as the `gdma` user, with the full execution environment of that user, and specify the GroundWork-supplied perl interpreter. The output is essentially the same for UNIX as for Windows.

There are several diagnostic steps that GDMA goes through each time it runs. Errors in the retrieval of the config file, or with a syntax error in that file, will be presented near the top of this output. Any problems running the checks will appear in the last sections.

- **On Windows** - Open a command shell, and type (Windows GDMA can be installed in alternate locations, you may need to enter a different path. For example in 64-bit Windows, GDMA is installed in `%Program Files (x86)\groundwork\gdma` by default).

Example poller output:

```
cd \Program Files (x86)\groundwork\gdma\bin
gdma_poll.exe -i -d2 -x
```

```
Getting http://10.0.10.109/gdma/gwmon_ip-10-0-10-219.cfg to C:\Program Files
(x86)\groundwork\gdma\config\gwmon_ip-10-0-10-219.cfg (timeout 10).
Attempting to get http://10.0.10.109/gdma/gwmon_ip-10-0-10-219.cfg to C:\Program
Files (x86)\groundwork\gdma\config\gwmon_ip-10-0-10-
219.cfg (timeout is set to 10 seconds).
Failed to get config file http://10.0.10.109/gdma/gwmon_ip-10-0-10-219.cfg -- 304
Not Modified
(304's are generally safe to ignore)
Poller is using hostname "ip-10-0-10-219".
Successfully fetched config file.
Checking config file "C:\Program Files
(x86)\groundwork\gdma\config\gdma_auto.conf" for a change.
Checking config file "C:\Program Files
(x86)\groundwork\gdma\config\gdma_override.conf" for a change.
Reloading config file.
Successfully read host config file.
Host config file syntax is ok.
Spooling poller startup message.
spool_startup_info: Poller 2.4.0 started at Friday April 7 12:35:21 2017
Pushed result into the buffer: 0 0 1491593721 ip-10-0-10-219 gdma_poller 0
Poller 2.4.0 started at Friday April 7 12:35:21 2017
Executing poller normal mode.
The configuration file contains:
Auto_Register_Attempts = fibonacci
Auto_Register_Host_Profile = gdma-windows-host
Auto_Register_Max_Interval = 86400
Auto_Register_Pass = gdma
Auto_Register_Service_Profile =
Auto_Register_User = gdma
Check_gdma_wmi_cpu[1]_Command = cscript.exe //nologo //T:60
'$Plugin_Directory$\v2\check_cpu_load_percentage.vbs' -h $Monitor_Host$ -inst
_Total -t 80,90
Check_gdma_wmi_cpu[1]_Enable = ON
Check_gdma_wmi_cpu[1]_Check_Interval = 1
```

```
Check_gdma_wmi_cpu[1]_Service = gdma_21_wmi_cpu
Check_gdma_wmi_disk_C[1]_Command = cscript.exe //nologo //T:60
'$Plugin_Directory$\v2\check_disks_percentage_space_used.vbs' -h $Monitor_Host$
-inst C: -t 80,90
Check_gdma_wmi_disk_C[1]_Enable = ON
Check_gdma_wmi_disk_C[1]_Check_Interval = 1
Check_gdma_wmi_disk_C[1]_Service = gdma_21_wmi_disk_C
Check_gdma_wmi_disk_all[1]_Command = cscript.exe //nologo //T:60
'$Plugin_Directory$\v2\check_disks_percentage_space_used.vbs' -h $Monitor_Host$
-inst * -t 80,90
Check_gdma_wmi_disk_all[1]_Enable = ON
Check_gdma_wmi_disk_all[1]_Check_Interval = 1
Check_gdma_wmi_disk_all[1]_Service = gdma_21_wmi_disk_all
Check_gdma_wmi_disktransfers[1]_Command = cscript.exe //nologo //T:60
'$Plugin_Directory$\v2\check_counter_counter.vbs' -h $Monitor_Host$ -class
Win32_PerfRawData_PerfDisk_PhysicalDisk -inst Name=_Total -prop
DiskTransfersPersec -w 10 -c 20
Check_gdma_wmi_disktransfers[1]_Enable = ON
Check_gdma_wmi_disktransfers[1]_Check_Interval = 1
Check_gdma_wmi_disktransfers[1]_Service = gdma_21_wmi_disktransfers
Check_gdma_wmi_mem[1]_Command = cscript.exe //nologo //T:60
'$Plugin_Directory$\v2\check_memory_percentage_space_used.vbs' -h $Monitor_Host$
-inst _Total -t 80,90
Check_gdma_wmi_mem[1]_Enable = ON
Check_gdma_wmi_mem[1]_Check_Interval = 1
Check_gdma_wmi_mem[1]_Service = gdma_21_wmi_mem
Check_gdma_wmi_memory_pages[1]_Command = cscript.exe //nologo //T:60
'$Plugin_Directory$\v2\check_counter_counter.vbs' -h $Monitor_Host$ -class
Win32_PerfRawData_PerfOS_Memory -inst * -prop PagesPerSec -w 10 -c 20
Check_gdma_wmi_memory_pages[1]_Enable = ON
Check_gdma_wmi_memory_pages[1]_Check_Interval = 1
Check_gdma_wmi_memory_pages[1]_Service = gdma_21_wmi_memory_pages
Check_gdma_wmi_uptime[1]_Command = cscript.exe //nologo //T:60
'$Plugin_Directory$\v2\get_system_uptime.vbs' -h 127.0.0.1 -w 1800 -c 900
Check_gdma_wmi_uptime[1]_Enable = ON
Check_gdma_wmi_uptime[1]_Check_Interval = 1
Check_gdma_wmi_uptime[1]_Service = gdma_21_wmi_uptime
ConfigFile_Pull_Cycle = 1
ConfigPull_Timeout = 10
Enable_Auto = On
Enable_Local_Logging = off
Forced_Hostname = ip-10-0-10-219
GDMAConfigDir = gdma
GDMA_Auto_Host = gdma-autohost
GDMA_Auto_Service = gdma_auto
GDMA_Multihost = off
HostSequenceNumber = 3
Logdir = C:\Program Files (x86)\groundwork\gdma\log\
Max_Concurrent_Hosts = 1
Max_Server_Redirects = 5
NumHostsInInstallation = 6
PluginPull_Timeout = 40
Poller_Plugin_Directory = C:\PROGRA~2\GROUND~1\gdma\libexec\
Poller_Plugin_Timeout = 20
Poller_Proc_Interval = 600
Poller_Pull_Failure_Interval = 86400
Poller_Service = gdma_poller
Poller_Status = On
Spooler_Batch_Size = 20
Spooler_Max_Retries = 10
Spooler_NSCA_Config = C:\Program Files (x86)\groundwork\gdma\config\send_nsca.cfg
Spooler_NSCA_Port = 5667
Spooler_NSCA_Program = C:\Program Files (x86)\groundwork\gdma\bin\send_nsca.exe
Spooler_NSCA_Timeout = 5
Spooler_Proc_Interval = 30
Spooler_Retention_Time = 900
Spooler_Service = gdma_spooler
Spooler_Status = Updates
Target_Server = http://10.0.10.109
Use_Long_Hostname = on
```

```
Executing gdma_run_checks; arguments are: -H "ip-10-0-10-219" -T 20 -p
"C:\Program Files (x86)\groundwork\gdma" -s "gdma_poller" -L
"C:\PROGRA~2\GROUND~1\gdma\libexec" -d 2 -i
run_normal_mode: OK Poller processed 7 checks in 4.134 secs. Using 0.69% of 600
sec Polling Interval. |NumChecks=7;;; TimeSecs=4.134;;;0; PctTime=0.69;60;80;0;
ExecTime=4.13421201705933;;;0;
Pushed result into the buffer: 0 0 1491593725 ip-10-0-10-219 gdma_poller 0
OK Poller processed 7 checks in 4.134 secs. Using 0.69% of 600 sec Polling
```

```
Interval.|NumChecks=7;;; TimeSecs=4.134;;;0; PctTime=0.69;60;80;0;
ExecTime=4.13421201705933;;;0;
Loop count=1. Last loop exec time = 4.139 seconds.
```

Example spooler output:

```
cd \Program Files (x86)\groundwork\gdma\bin
gdma_spool_processor.exe -i -d2 -x
```

```
Spooler is using hostname "ip-10-0-10-219".
Checking config file "C:\Program Files
(x86)\groundwork\gdma\config\gdma_auto.conf" for a change.
Checking config file "C:\Program Files
(x86)\groundwork\gdma\config\gdma_override.conf" for a change.
Checking config file "C:\Program Files
(x86)\groundwork\gdma\config\gmon_ip-10-0-10-219.cfg" for a change.
Reloading config files.
Successfully read host config file.
Host config file syntax is ok.
The configuration file contains:
Auto_Register_Attempts = fibonacci
Auto_Register_Host_Profile = gdma-windows-host
Auto_Register_Max_Interval = 86400
Auto_Register_Pass = gdma
Auto_Register_Service_Profile =
Auto_Register_User = gdma
Check_gdma_wmi_cpu[1]_Command = cscript.exe //nologo //T:60
'$Plugin_Directory$\v2\check_cpu_load_percentage.vbs' -h $Monitor_Host$ -inst
_Total -t 80,90
Check_gdma_wmi_cpu[1]_Enable = ON
Check_gdma_wmi_cpu[1]_Check_Interval = 1
Check_gdma_wmi_cpu[1]_Service = gdma_21_wmi_cpu
Check_gdma_wmi_disk_C[1]_Command = cscript.exe //nologo //T:60
'$Plugin_Directory$\v2\check_disks_percentage_space_used.vbs' -h $Monitor_Host$
-inst C: -t 80,90
Check_gdma_wmi_disk_C[1]_Enable = ON
Check_gdma_wmi_disk_C[1]_Check_Interval = 1
Check_gdma_wmi_disk_C[1]_Service = gdma_21_wmi_disk_C
Check_gdma_wmi_disk_all[1]_Command = cscript.exe //nologo //T:60
'$Plugin_Directory$\v2\check_disks_percentage_space_used.vbs' -h $Monitor_Host$
-inst * -t 80,90
Check_gdma_wmi_disk_all[1]_Enable = ON
Check_gdma_wmi_disk_all[1]_Check_Interval = 1
Check_gdma_wmi_disk_all[1]_Service = gdma_21_wmi_disk_all
Check_gdma_wmi_disktransfers[1]_Command = cscript.exe //nologo //T:60
'$Plugin_Directory$\v2\check_counter_counter.vbs' -h $Monitor_Host$ -class
Win32_PerfRawData_PerfDisk_PhysicalDisk -inst Name=_Total -prop
DiskTransfersPerSec -w 10 -c 20
Check_gdma_wmi_disktransfers[1]_Enable = ON
Check_gdma_wmi_disktransfers[1]_Check_Interval = 1
Check_gdma_wmi_disktransfers[1]_Service = gdma_21_wmi_disktransfers
Check_gdma_wmi_mem[1]_Command = cscript.exe //nologo //T:60
'$Plugin_Directory$\v2\check_memory_percentage_space_used.vbs' -h $Monitor_Host$
-inst _Total -t 80,90
Check_gdma_wmi_mem[1]_Enable = ON
Check_gdma_wmi_mem[1]_Check_Interval = 1
Check_gdma_wmi_mem[1]_Service = gdma_21_wmi_mem
Check_gdma_wmi_memory_pages[1]_Command = cscript.exe //nologo //T:60
'$Plugin_Directory$\v2\check_counter_counter.vbs' -h $Monitor_Host$ -class
Win32_PerfRawData_PerfOS_Memory -inst * -prop PagesPerSec -w 10 -c 20
Check_gdma_wmi_memory_pages[1]_Enable = ON
Check_gdma_wmi_memory_pages[1]_Check_Interval = 1
Check_gdma_wmi_memory_pages[1]_Service = gdma_21_wmi_memory_pages
Check_gdma_wmi_uptime[1]_Command = cscript.exe //nologo //T:60
'$Plugin_Directory$\v2\get_system_uptime.vbs' -h 127.0.0.1 -w 1800 -c 900
Check_gdma_wmi_uptime[1]_Enable = ON
```

```
Check_gdma_wmi_uptime[1]_Check_Interval = 1
Check_gdma_wmi_uptime[1]_Service = gdma_21_wmi_uptime
ConfigFile_Pull_Cycle = 1
ConfigPull_Timeout = 10
Enable_Auto = off
Enable_Local_Logging = off
Forced_Hostname = ip-10-0-10-219
GDMAConfigDir = gdma
GDMA_Auto_Host = gdma-autohost
GDMA_Auto_Service = gdma_auto
GDMA_Multihost = off
HostSequenceNumber = 3
Logdir = C:\Program Files (x86)\groundwork\gdma\log\
Max_Concurrent_Hosts = 1
Max_Server_Redirects = 5
NumHostsInInstallation = 6
PluginPull_Timeout = 40
Poller_Plugin_Directory = C:\Program Files (x86)\groundwork\gdma\libexec\
Poller_Plugin_Timeout = 20
Poller_Proc_Interval = 600
Poller_Pull_Failure_Interval = 86400
Poller_Service = gdma_poller
Poller_Status = On
Spooler_Batch_Size = 20
Spooler_Max_Retries = 10
Spooler_NSCA_Config = C:\Program Files (x86)\groundwork\gdma\config\send_nsca.cfg
Spooler_NSCA_Port = 5667
Spooler_NSCA_Program = C:\Program Files (x86)\groundwork\gdma\bin\send_nsca.exe
Spooler_NSCA_Timeout = 5
Spooler_Proc_Interval = 30
Spooler_Retention_Time = 900
Spooler_Service = gdma_spooler
Spooler_Status = Updates
Target_Server = http://10.0.10.109
Use_Long_Hostname = on
Spooler is using hostname "ip-10-0-10-219".
Spooling spool processor startup message.
spool_startup_info:: Spool processor 2.4.0 started at Friday April 7 13:21:01 2017
Pushed result into the buffer: 0 0 1491596461 ip-10-0-10-219 gdma_spooler 0
Spool processor 2.4.0 started at Friday April 7 13:21:01 2017
Spool processor running in normal mode.
Processing live targets.
Processing for target 10.0.10.109
Spooling heart beat message.
Spooler heart beat message: Spooler transmitted 1 results in 0.052 secs |
RESULTS=1;;; TIME=0.052s;;;
```

```
Pushed result into the buffer: 0 0 1491596461 ip-10-0-10-219 gdma_spooler 0
Spooler transmitted 1 results in 0.052 secs | RESULTS=1;;; TIME=0.052s;;;
Loop count = 1. Last loop exec time = 0.053 seconds.
```

- **On Linux and AIX**

Example poller output:

```
su - gdma
cd /usr/local/groundwork/gdma/bin
/usr/local/groundwork/perl/bin/perl gdma_poll.pl -i -d2 -x
```

```
Getting http://10.0.10.109/gdma/gwmon_ip-10-0-10-209.localdomain.cfg to
/usr/local/groundwork/gdma/config/gwmon_ip-10-0-10-209.localdomain.cfg (timeout
10).
Attempting to get http://10.0.10.109/gdma/gwmon_ip-10-0-10-209.localdomain.cfg to
/usr/local/groundwork/gdma/config/gwmon_ip-10-0-10-209.localdomain.cfg (timeout is
set to 10 seconds).
Failed to get config file
http://10.0.10.109/gdma/gwmon_ip-10-0-10-209.localdomain.cfg -- 304 Not Modified
(304's are generally safe to ignore)
Poller is using hostname "ip-10-0-10-209.localdomain".
Successfully fetched config file.
Checking config file "/usr/local/groundwork/gdma/config/gdma_auto.conf" for a
change.
Checking config file "/usr/local/groundwork/gdma/config/gdma_override.conf" for a
change.
Reloading config file.
Successfully read host config file.
Host config file syntax is ok.
Spooling poller startup message.
spool_startup_info: Poller 2.4.0 started at Friday April 7 11:38:45 2017
[Pushed result into the buffer: 0 0 1491590325 ip-10-0-10-209.localdomain
gdma_poller 0 Poller 2.4.0 started at Friday April 7 11:38:45 2017
Executing poller normal mode.
The configuration file contains:
Auto_Register_Attempts = fibonacci
Auto_Register_Host_Profile = gdma-linux-host
Auto_Register_Max_Interval = 86400
Auto_Register_Pass = gdma
Auto_Register_Service_Profile =
Auto_Register_User = gdma
Check_gdma_linux_disk_root[1]_Command = check_disk -w 10% -c 5% -t 10 -p
Check_gdma_linux_disk_root[1]_Enable = ON
Check_gdma_linux_disk_root[1]_Check_Interval = 1
Check_gdma_linux_disk_root[1]_Service = linux_disk_root
Check_gdma_linux_load[1]_Command = check_load -r -w 90,75,60 -c 100,95,90
Check_gdma_linux_load[1]_Enable = ON
Check_gdma_linux_load[1]_Check_Interval = 1
Check_gdma_linux_load[1]_Service = linux_load
Check_gdma_linux_mem[1]_Command = check_mem.pl -f -w 20 -c 10
Check_gdma_linux_mem[1]_Enable = ON
Check_gdma_linux_mem[1]_Check_Interval = 1
Check_gdma_linux_mem[1]_Service = linux_mem
Check_gdma_linux_process_count[1]_Command = check_procs -w 80 -c 100
Check_gdma_linux_process_count[1]_Enable = ON
Check_gdma_linux_process_count[1]_Check_Interval = 1
Check_gdma_linux_process_count[1]_Service = linux_process_count
Check_gdma_linux_swap[1]_Command = check_swap -w 10% -c 5%
Check_gdma_linux_swap[1]_Enable = ON
Check_gdma_linux_swap[1]_Check_Interval = 1
Check_gdma_linux_swap[1]_Service = linux_swap
Check_gdma_linux_uptime[1]_Command = check_system_uptime.pl -w 1800 -c 900
Check_gdma_linux_uptime[1]_Enable = ON
Check_gdma_linux_uptime[1]_Check_Interval = 1
```

```
Check_gdma_linux_uptime[1]_Service = linux_uptime
ConfigFile_Pull_Cycle = 1
ConfigPull_Timeout = 10
Enable_Auto = On
Enable_Local_Logging = on
Forced_Hostname = ip-10-0-10-209.localdomain
GDMAConfigDir = gdma
GDMA_Auto_Host = gdma-autohost
GDMA_Auto_Service = gdma_auto
GDMA_Multihost = off
HostSequenceNumber = 5
Logdir = /usr/local/groundwork/gdma/log/
Max_Concurrent_Hosts = 1
Max_Server_Redirects = 5
NumHostsInInstallation = 6
PluginPull_Timeout = 40
Poller_Plugin_Directory = /usr/local/groundwork/nagios/libexec
Poller_Plugin_Timeout = 5
Poller_Proc_Interval = 600
Poller_Pull_Failure_Interval = 86400
Poller_Service = gdma_poller
Poller_Status = On
Spooler_Batch_Size = 20
Spooler_Max_Retries = 10
Spooler_NSCA_Config = /usr/local/groundwork/gdma/config/send_nsca.cfg
Spooler_NSCA_Port = 5667
Spooler_NSCA_Program = /usr/local/groundwork/common/bin/send_nsca
Spooler_NSCA_Timeout = 5
Spooler_Proc_Interval = 30
Spooler_Retention_Time = 900
Spooler_Service = gdma_spooler
Spooler_Status = Updates
Target_Server = http://10.0.10.109
Use_Long_Hostname = on
Executing gdma_run_checks; arguments are: -H ip-10-0-10-209.localdomain -T 5 -p
/usr/local/groundwork/gdma -s gdma_poller -L /usr/local/groundwork/nagios/libexec
-l -D /usr/local/groundwork/gdma/log -d 2 -i
Check Check_gdma_linux_disk_root ran 1 iterations ago; check will run now.
Executing 'Check_gdma_linux_disk_root' - iteration 1
Check_gdma_linux_disk_root[1]_Command = check_disk -w 10% -c 5% -t 10 -p
Check_gdma_linux_disk_root[1]_Enable = ON
Check_gdma_linux_disk_root[1]_Check_Interval = 1
Check_gdma_linux_disk_root[1]_Service = linux_disk_root
PLUGIN COMMAND STRING: check_disk -w 10% -c 5% -t 10 -p
Executing plugin command on linux
Wait for child process completed at Fri Apr 7 11:38:45 2017.
Plugin process terminated with code 0.
Raw plugin output was:
Executing command (check_disk -w 10% -c 5% -t 10 -p) DISK OK - free space: 3276 MB
(57% inode=85%);| =2463MB;5442;5744;0;6047
PLUGIN RESULTS: exit=0; text=[Fri Apr 7 11:38:45 2017] Executing command
(check_disk -w 10% -c 5% -t 10 -p) DISK OK - free space: 3276 MB (57% inode=85%);|
=2463MB;5442;5744;0;6047
Pushed result into the buffer: 0 0 1491590325 ip-10-0-10-209.localdomain
linux_disk_root 0 [Fri Apr 7 11:38:45 2017] Executing command (check_disk -w 10%
-c 5% -t 10 -p) DISK OK - free space: 3276 MB (57% inode=85%);|
=2463MB;5442;5744;0;6047
Check Check_gdma_linux_load ran 1 iterations ago; check will run now.
Executing 'Check_gdma_linux_load' - iteration 1
Check_gdma_linux_load[1]_Command = check_load -r -w 90,75,60 -c 100,95,90
Check_gdma_linux_load[1]_Enable = ON
Check_gdma_linux_load[1]_Check_Interval = 1
Check_gdma_linux_load[1]_Service = linux_load
PLUGIN COMMAND STRING: check_load -r -w 90,75,60 -c 100,95,90
Executing plugin command on linux
Wait for child process completed at Fri Apr 7 11:38:45 2017.
Plugin process terminated with code 0.
Raw plugin output was:
Executing command (check_load -r -w 90,75,60 -c 100,95,90) OK - load average:
0.00, 0.00, 0.00|load1=0.000;90.000;100.000;0; load5=0.000;75.000;95.000;0;
```

```
load15=0.000;60.000;90.000;0;
PLUGIN RESULTS:  exit=0; text=[Fri Apr 7 11:38:45 2017] Executing command
(check_load -r -w 90,75,60 -c 100,95,90) OK - load average: 0.00, 0.00,
0.00|load1=0.000;90.000;100.000;0; load5=0.000;75.000;95.000;0;
load15=0.000;60.000;90.000;0;
Pushed result into the buffer:  0 0 1491590325 ip-10-0-10-209.localdomain
linux_load 0 [Fri Apr 7 11:38:45 2017] Executing command (check_load -r -w
90,75,60 -c 100,95,90) OK - load average: 0.00, 0.00,
```

```
0.00|load1=0.000;90.000;100.000;0; load5=0.000;75.000;95.000;0;
load15=0.000;60.000;90.000;0;
...
```

Example spooler output:

```
su - gdma
cd /usr/local/groundwork/gdma/bin
/usr/local/groundwork/perl/bin/perl gdma_poll.pl -i -d2 -x
```

```
Spooler is using hostname "ip-10-0-10-209.localdomain".
Checking config file "/usr/local/groundwork/gdma/config/gdma_auto.conf" for a
change.
Checking config file "/usr/local/groundwork/gdma/config/gdma_override.conf" for a
change.
Checking config file
"/usr/local/groundwork/gdma/config/gwmon_ip-10-0-10-209.localdomain.cfg" for a
change.
Reloading config files.
Successfully read host config file.
Host config file syntax is ok.
The configuration file contains:
Auto_Register_Attempts = fibonacci
Auto_Register_Host_Profile = gdma-linux-host
Auto_Register_Max_Interval = 86400
Auto_Register_Pass = gdma
Auto_Register_Service_Profile =
Auto_Register_User = gdma
Check_gdma_linux_disk_root[1]_Command = check_disk -w 10% -c 5% -t 10 -p
Check_gdma_linux_disk_root[1]_Enable = ON
Check_gdma_linux_disk_root[1]_Check_Interval = 1
Check_gdma_linux_disk_root[1]_Service = linux_disk_root
Check_gdma_linux_load[1]_Command = check_load -r -w 90,75,60 -c 100,95,90
Check_gdma_linux_load[1]_Enable = ON
Check_gdma_linux_load[1]_Check_Interval = 1
Check_gdma_linux_load[1]_Service = linux_load
Check_gdma_linux_mem[1]_Command = check_mem.pl -f -w 20 -c 10
Check_gdma_linux_mem[1]_Enable = ON
Check_gdma_linux_mem[1]_Check_Interval = 1
Check_gdma_linux_mem[1]_Service = linux_mem
Check_gdma_linux_process_count[1]_Command = check_procs -w 80 -c 100
Check_gdma_linux_process_count[1]_Enable = ON
Check_gdma_linux_process_count[1]_Check_Interval = 1
Check_gdma_linux_process_count[1]_Service = linux_process_count
Check_gdma_linux_swap[1]_Command = check_swap -w 10% -c 5%
Check_gdma_linux_swap[1]_Enable = ON
Check_gdma_linux_swap[1]_Check_Interval = 1
Check_gdma_linux_swap[1]_Service = linux_swap
Check_gdma_linux_uptime[1]_Command = check_system_uptime.pl -w 1800 -c 900
Check_gdma_linux_uptime[1]_Enable = ON
Check_gdma_linux_uptime[1]_Check_Interval = 1
Check_gdma_linux_uptime[1]_Service = linux_uptime
ConfigFile_Pull_Cycle = 1
ConfigPull_Timeout = 10
Enable_Auto = off
Enable_Local_Logging = on
Forced_Hostname = ip-10-0-10-209.localdomain
GDMAConfigDir = gdma
GDMA_Auto_Host = gdma-autohost
GDMA_Auto_Service = gdma_auto
GDMA_Multihost = off
HostSequenceNumber = 5
Logdir = /usr/local/groundwork/gdma/log/
Max_Concurrent_Hosts = 1
Max_Server_Redirects = 5
NumHostsInInstallation = 6
```

```
PluginPull_Timeout = 40
Poller_Plugin_Directory = /usr/local/groundwork/nagios/libexec
Poller_Plugin_Timeout = 5
Poller_Proc_Interval = 600
Poller_Pull_Failure_Interval = 86400
Poller_Service = gdma_poller
Poller_Status = On
Spooler_Batch_Size = 20
Spooler_Max_Retries = 10
Spooler_NSCA_Config = /usr/local/groundwork/gdma/config/send_nsca.cfg
Spooler_NSCA_Port = 5667
Spooler_NSCA_Program = /usr/local/groundwork/common/bin/send_nsca
Spooler_NSCA_Timeout = 5
Spooler_Proc_Interval = 30
Spooler_Retention_Time = 900
Spooler_Service = gdma_spooler
Spooler_Status = Updates
Target_Server = http://10.0.10.109
Use_Long_Hostname = on
Spooler is using hostname "ip-10-0-10-209.localdomain".
Spooling spool processor startup message.
spool_startup_info: Spool processor 2.4.0 started at Friday April 7 12:13:33 2017
Pushed result into the buffer: 0 0 1491592413 ip-10-0-10-209.localdomain
gdma_spooler 0 Spool processor 2.4.0 started at Friday April 7 12:13:33 2017
Spool processor running in normal mode.
Processing live targets.
Processing for target 10.0.10.109
Wait for child process completed at 1491592413.
=====
send_nsca output:
-----
/usr/local/groundwork/common/bin/send_nsca -H 10.0.10.109 -p 5667 -od -c
/usr/local/groundwork/gdma/config/send_nsca.cfg -to 5
Invoked send_nsca successfully ...
1 data packet(s) sent to host successfully.
=====
Spooling heart beat message.
Spooler heart beat message: Spooler transmitted 1 results in 0.017 secs |
RESULTS=1;;; TIME=0.017s;;;
Pushed result into the buffer: 0 0 1491592413 ip-10-0-10-209.localdomain
```

```
gdma_spooler 0 Spooler transmitted 1 results in 0.017 secs | RESULTS=1;;;  
TIME=0.017s;;;  
Loop count = 1. Last loop exec time = 0.017 seconds.
```

- **On Solaris**

```
su - gdma  
cd /opt/groundwork/gdma/bin  
/opt/groundwork/perl/bin/perl gdma_poll.pl -i -d2 -x
```