

GDMA FAQs

GDMA FAQs

Q: How can I speed up auto-registration?

- **A:** Regarding auto-registration processing, if an auto-registration request doesn't yield any positive results, the poller won't keep banging on the server every iteration, it waits for at least an hour before trying again as the interval is currently hardcoded. To speed things up for testing, especially to avoid the initial 10-minute wait before the first auto-registration attempt after the poller starts and finds it has no local copy of the host configuration file, you will probably want to set the processing interval to 60 (which is one minute). To change the directive go to *Configuration > Hosts > Host externals > Modify > {gdma-linux} > Poller_Proc_Interval = "60"*. If you do change this value for testing, be sure to change it back (typically to 600) for production use. If auto-registration fails on a given attempt and you don't want to wait the extra hour for the following attempt, just bounce GDMA on the client machine, the [DOCDEV:How to stop and start services] page describes the various platform-specific commands used to start and stop the GDMA client.

Q: What is the role of profiles in Auto Registration?

- **A:** When the client makes an auto-registration call to the server, it passes along a small amount of information about the client system, including the configured `Auto_Register_Host_Profile` and `Auto_Register_Service_Profile` values. The `foundation/scripts/registerAgentByProfile.pl` script on the server is called to validate the host attributes, add the host to Monarch, build externals for the host (based on whatever externals were attached to the host profile, or to the services attached to the service profile(s)), and return the final hostname. Building externals means that the associated host and service externals are collected together, processed via possible macro substitution, and written to a location where the client can pick them up.

The host profiles will form the basis for configuring new auto-registered GDMA hosts running the respective operating-system types. You are not required to use these profiles; the GDMA client installer allows you to select an alternate host profile and an optional service profile to be applied on the server to the auto-registered client's configuration. Also see the `default_host_profile` setting in the `/usr/local/groundwork/config/register_agent.properties` file.

Q: What is the actual Auto Registration process?

- **A:** In practice, a green system operates in the following way. An un-configured poller will reach for a local copy of the host-config file, find none, and go to sleep for 10 minutes (configured in `gdma_auto.conf` as the `Poller_Proc_Interval`). On waking up, when it again sees it still cannot find its config file, if it has auto-registration credentials, it will do the auto-registration thing; otherwise, it will fall back to the old auto-configure protocol. All the auto-registration thing does is to make a call to the server, and look at the results. If the results include a hostname, the poller stores the hostname in the `gdma_override.conf` file. At that moment, it still doesn't have a host config file in hand. On the next cycle, it should pull the externals file from the GroundWork server, save it locally, and start monitoring the client machine.

The spooler, in its own cycle time, eventually sees that the configuration files have changed, and re-reads them. Then it will see the forced hostname and adapt to it.

When an auto-registration request comes into the server, it first tries to look up in Monarch to see if the host is already registered. There is a special Perl package installed (`foundation/scripts/AutoRegistration.pm`) that implements a pass of possible host-attribute recoding before this lookup. If that lookup fails, a second possible recoding (presumably with different logic this time) and lookup is attempted. If both passes fail and the package has not concluded that the attributes are totally invalid, the host will be added to Monarch.

The second pass of recoding allows IP = hostname lookups based on specific addresses listed in the server `config/register_agent.properties` file, for cases where DNS just isn't operating right on the client and there's no way to fix that easily.



Auto-registered hosts will begin monitoring as soon as they receive their own host configuration file from the server. They will then start sending on the results of those checks to the GroundWork server. At this point, however, Nagios won't know of the existence of such hosts, and it will casually drop those results on the floor, probably with a notice to that effect in the `nagios/var/nagios.log` file. Results from such hosts will continue to be ignored until the administrator (or perhaps some automated process) runs a Commit operation.

Q: Which GroundWork server is the auto-registration request sent?

- **A:** It's the server mentioned as the first value of the `Target_Server` option in the `config/gdma_auto.conf` file on the client.

Q: What happens if the GDMA client is already present?

- **A:** During auto-registration, if the GDMA client machine is already present in Monarch and doesn't need to be freshly added to the

configuration, the server makes up to two separate attempts to look up the host within Monarch, to find it using possibly different host attributes. If one of those attempts succeeds, the existing hostname in Monarch will be returned to the client as its official, final hostname. But even in this case, the server attempts to ensure that the desired host profile, service profile, hostgroup, and Monarch configuration group assignments are made to this host.

Q: What can I do if DNS is not working at the client?

- **A:** My GDMA client cannot reliably determine its own hostname to send to the server, because DNS is not working at the client. Many failures of this type can be corrected by the server during the auto-registration process. Logic in the `AutoRegistration.pm` module looks at the IP address sent by the client as well, and tries to determine a correct match between the IP address and the hostname before settling on a final hostname to add to Monarch and to publish back to the client machine. See also the `<hardcoded_hostnames>` option in the `config/register_agent.properties` file, for handling particularly pernicious problems.

Q: Can I change the form of the hostname the client gets registered under?

- **A:** An out-of-the-box setup will cause auto-registered hosts to have fully qualified hostnames within GroundWork Monitor. This form was chosen to provide the least chance of ambiguity in large installations, especially those that serve multiple domains that might otherwise have hostname collisions. But it is not a requirement of operation. The client tries to determine its own hostname using standard heuristics, and it sends that hostname to the server as part of the auto-registration request. The server may or may not accept that hostname as-is, but in any case, the server's decision is final. There are two controls, and possibly other settings, to play with.
 - On the client side, you can set the `Forced_Hostname` option in the `config/gdma_auto.conf` file. See [GDMA Advanced](#).



- With auto-registration in play, this option in the `gdma_auto.conf` file really only affects the name passed to the server by the client during its auto-registration attempt. Afterward, the value of this option dynamically saved by the client in the `gdma_override.conf` file will rule.
- The `Use_Long_Hostname` option will have no practical effect when auto-registration is in effect. The client always attempts to send a fully-qualified hostname to the server as part of an auto-registration attempt, to give the server the best chance of accurately identifying the client.

- On the server side, you can adjust the `hostname_qualification` option in the `config/register_agent.properties` file, to globally select long or short names, or to pass the decision to your own custom logic.



- If the hostname is being determined by the `<hardcoded_hostnames>` section of the `config/register_agent.properties` file, that will be the place you will need to change it.

- In the extreme, if you need some kind of specialized hostname determination that is not covered by the options above, you can modify the logic by which the server chooses the client hostname. You might do this, for instance, if you want to use unqualified hostnames for most of your machines, and fully qualified hostnames only for machines on particular subnets. The client sends multiple attributes to the server as part of an auto-registration request: the type of agent making the request (e.g., *GDMA*), its own hostname (as best it knows it), its IP address, its MAC address, and the generic type of operating system running on the host. Custom logic can use any or all of these attributes to make a final determination of the client hostname. To implement custom server logic in this area, you will need to:
 - Copy the `foundation/scripts/AutoRegistration.pm` file into some sibling file of your own name (e.g., `AcmeCorporationAutoReg.pm`).
 - Change the `package` declaration in your file to reflect that name:

```
package AcmeCorporationAutoReg;
```

- Change the logic in the `soft_recode_host_attributes()` and/or `hard_recode_host_attributes()` routines as you need.
- Edit the documentation in the file at the end of your own version of this package to reflect your logic changes.
- Change the `customer_network_package` option in the `config/register_agent.properties` file to name your package instead of the `AutoRegistration` package.

```
customer_network_package = "AcmeCorporationAutoReg"
```

- This last step will put your package into play.



- GroundWork Monitor will overwrite the `AutoRegistration.pm` file in future releases, which is part of why you are advised to change the filename and package name if you change the content.
- You should be aware that both client-side GDMA and the server-side scripting always lowercase whatever hostname they use, to provide consistency in the face of potential confusion. This behavior is not configurable, as it just causes too much headache to loosen up on this convention.

Q: Can I change the form of the hostname the client has been registered under?

- **A:** You can rename the host within Monarch (*Configuration > Hosts > hostgroup > host > Detail > Rename*). After doing so, go to *Configuration > Control > Build externals*, to rebuild the externals file using the modified hostname. After some time, the GDMA client will recognize that its externals file is no longer available on the server under the old name, and that will eventually cause it to exercise the auto-registration logic. At that time, the client should be recognized by the auto-registration request as being already in Monarch (given that the IP address has not changed, even though the hostname did), and the new name will be returned to the client for further operations.
 - If need be, you can speed up this process by bouncing GDMA on the client after the externals are rebuilt on the server, see the [DOCDEV:How to stop and start services] page describes the various platform-specific commands used to start and stop the GDMA client. It will take about 10 minutes (one `Poller_Proc_Interval`) for the auto-registration to occur on the client.
 - Of course, you will need to run a Commit operation (*Configuration > Control > Commit*) for Nagios to understand that the hostname has changed, so it doesn't start dropping the data for this host on the floor.
 - Renaming an existing host like this is subject to a caveat, namely that in the current release of GroundWork Monitor, performance data collected under the old hostname will not be transferred to the new hostname. The old data still exists, but it is stored in RRD files using the old hostname, while new performance data will be stored in RRD files using the new hostname. We don't currently have a utility to merge the old data into the new RRD files.

Q: Can I choose a different name for Auto-Registration and auto-registration?

- **A:** Yes. You can choose a different name for both the host group *Auto-Registration* and for the Monarch group *auto-registration* if you want, although you will need to modify the following settings in the `config/register_agent.properties` file:

```
default_hostgroup = "Auto-Registration"
```

```
default_monarch_group = "auto-registration"
```

```
# The default hostgroup which is passed to the hostgroups_to_assign()
# routine in the customer_network_package (configured below) for
# assigning to any host processed by the auto-registration code. That
# routine is free to override or ignore this default, depending on the
# full set of conditions that apply to the host.
default_hostgroup = "Auto-Registration"

# The default Monarch configuration group which is passed to the
# monarch_groups_to_assign() routine in the customer_network_package
# (configured below) for assigning to any host processed by the
# auto-registration code. That routine is free to override or ignore
# this default, depending on the full set of conditions that apply to
# the host.
default_monarch_group = "auto-registration"
```

Q: What configuration objects are associated with an auto-registered host?

- **A:** Multiple connections will be made during auto-registration:
 - Common practice is to have the host profile specified by the client be assigned and applied to the host. If the client does not specify a host profile, the server will assign a host profile based on the `default_host_profile` option in the `config/register_agent.properties` file. The server can modify these standard choices via the `host_profile_to_assign()` routine in a customized version of the `AutoRegistration.pm` module.
 - Any service profiles associated with the applied host profile will be applied to the host.
 - The client can optionally specify one additional service profile to be applied to the host.
 - Host and service externals from the host profile and from the services applied to the host will be copied to the configuration for the host. This is a critical link in how the individual GDMA options and service checks for the host are made known to the client to run its monitoring commands.
 - The host will become a member of the hostgroup listed as `default_hostgroup` in the

`config/register_agent.properties` file. If a more-complex setup is required, a customized version of the `AutoRegistration.pm` module can be used to return a list of host groups in the result of the `hostgroups_to_assign()` routine. If you truly don't want any auto-registered host to be a member of any hostgroup (probably not a practical setup), that can also be accomplished by having your custom version of the `AutoRegistration.pm` module have no `hostgroups_to_assign()` routine.

- If the host is not already a member of some Monarch configuration group, the host will become a member of the Monarch configuration group listed as `default_monarch_group` in the `config/register_agent.properties` file. This standard behavior can be modified via a customized version of the `AutoRegistration.pm` module, where the `monarch_groups_to_assign()` routine can be used to determine a list of Monarch groups to which the host should be assigned.
- Also see the `assign_monarch_groups_to_existing_group_hosts` option in the `config/register_agent.properties` file, if you need to allow a host which is already assigned to some Monarch configuration group to also be assigned to the Monarch configuration group(s) determined by the auto-registration logic.



The host must be assigned to at least one Monarch group in order to have externals built and published on the server, where the client can subsequently find them. Some customers might want to create and distribute the client host-configuration files via some other mechanism. A future version of GDMA will include the ability to assign an empty string to the `default_monarch_group` option in the `config/register_agent.properties` file, to suppress the assignment of the host to a Monarch group, for all hosts. In the meantime, the same effect can be achieved in a custom version of the `AutoRegistration.pm` module by either having the `monarch_groups_to_assign()` routine return an empty list of Monarch groups, or having no `monarch_groups_to_assign()` routine in your package.

- Administration of the Monarch configuration group can be made easier by manually assigning at least one hostgroup the host will be a member of to the Monarch group. Then all hosts that are auto-registered and that are a member of such a hostgroup won't have to be directly assigned to the Monarch group; the indirect association will suffice. This is generally the preferred setup, as it keeps the configuration much simpler.

Q: How do I customize the set of checks used by GDMA clients?

- **A:** GroundWork's standard profiles provide a basic set of monitoring checks for each platform (type of operating system). You can use custom profiles instead of the standard profiles that GroundWork provides and you can adjust the sets of checks however you like, selecting the sets of host and service checks you are most interested in.

Q: How do I get check definitions so they apply across many hosts?

- **A:** My monitoring-check definitions need to be written in a generic form so they apply across many hosts. But then how do I get them automatically customized for each particular host? Host and service externals can reference several different types of macros, and have those references be expanded when the externals files are built. Macros are currently expanded in the following order:
 - Nagios user-resource macros (the macros named as `USER1` through `USER32` in the *Configuration > Control > Nagios resource macros* screen).
 - Monarch configuration group macros (created in the *Configuration > Groups > Macros* screen, and associated with groups and possibly redefined in the *Configuration > Groups > Groups > groupname > Detail > Macros* screen). Since a host can be a member of multiple configuration groups arranged in parent-group/sub-group hierarchies, a standard order has to be defined in which such macros are applied. The rule is that Monarch configuration group macros are substituted in ascending order from subgroups to parent groups associated with the given host. This allows the most-specific value of a given macro (generally associated with a subgroup) to be used in preference to a more-general value of the same macro name (generally associated with a parent group, and used as a default value).
 - Certain standard macros: `HOSTNAME`, `HOSTADDRESS`, and `HOSTALIAS`.

In all of these cases, the macro references are recognized by the presence of both leading and trailing `$` characters around the macro names. So they would appear in your raw externals as references such as shown below. Lots of capabilities can be enabled and customized via this mechanism. GroundWork Professional Services can be engaged to help with complex requirements.

```
$USER32$  
$MYGROUPMACRO$  
$HOSTADDRESS$
```

Q: How do I change the set of checks for a host being monitored by GDMA?

- **A:** Modify the monitoring checks as you would for any other host — for instance, adding new service checks. Just make sure that each new host-service has associated externals applied to the host-service by the time you are done with your modifications, to define things like the command to run on the GDMA host. That way, the GDMA host will find out what it needs to do to perform the desired monitoring. Once your changes are done, build externals (to make the changes available to the GDMA host), then run a Commit (to let Nagios know that the set of configured checks has changed). Soon thereafter, the GDMA host will discover that its cached copy of the externals file is out-of-date, and it will download a fresh copy and begin monitoring according to the new setup. About 5 minutes after the Commit operation (per the setting of the Service performance data file processing interval, on page 3 of the *Control > Nagios main configuration*),

Nagios will start the performance-processing script. Performance data will start to be processed again, which will pick up any new checks for the host, create new RRD files if necessary, and make graphs for new metrics available in Status Viewer.

Q: How can I disable Automated Agent Registration on a GDMA client machine?

- **A:** Disabling auto-registration for an individual host can be effected on the client side by editing the `config/gdma_auto.conf` file and either setting the auto-registration access credentials to empty strings, or just commenting out those lines in the file. Such a setup will instead invoke the older auto-configure protocol if the client cannot find its own host-configuration file on the server.

```
Auto_Register_Pass = ""
Auto_Register_User = ""
```

Q: How can I completely disable Automated Agent Registration?

- **A:** To stop all Automated Agent Registration requests from succeeding you can disable the processing.
 - In the `/usr/local/groundwork/config/register_agent.properties` file, you can set:

```
enable_processing = no
```

Example default configuration:

```
# =====
# GroundWork Distributed Agent Registration Configuration Parameters
# =====

# Whether to process anything. Turn this off if you want to disable this
# process completely, so auto-registration is prohibited. This option is
# turned on in the default configuration file because the standard values
# are reasonable for most installations, so we don't feel it is necessary
# to block auto-registration until it is locally configured. To get the
# software to run, it must be turned on here once the rest of the setup is
# correct for your installation.
# [yes/no]
enable_processing = yes
```

Q: How do I uninstall GDMA?

- **A:** To uninstall GDMA:
 - **For Linux:**
 - Access a command line interface for the Linux GDMA client server, (e.g. PuTTY).
 - Backup any custom plugins in the `/usr/local/groundwork` directory structure, you may have added. You will need to move these to the new location.
 - Next, run the following script to uninstall the agent:

```
/usr/local/groundwork/uninstall
```

- You will be prompted with the question *Do you want to uninstall GroundWork Distributed Monitoring Agent and all of its modules?*, confirm the uninstall.
- Remove the `groundwork` sub-directory and all sub-directories:

```
rm -Rf /usr/local/groundwork
```

- On the GroundWork server delete the GDMA client host within the *Auto Registration* host group (*Configuration > Hosts > Hosts > Auto-Registration > <hostname> > Detail > delete*), and run a *Commit* operation (*Configuration > Control > Commit*).
- **For Windows:**
 - Access the File Manager for the Windows GDMA client server, (e.g. Start, Computer).
 - Backup any custom plugins in the `C:\Program Files (x86)\groundwork` directory structure, you may have added. You will need to move these to the new location.
 - Next, from the `C:\Program Files (x86)\groundwork` directory, select and run the `uninstall.exe` file. You will be prompted with the question *Do you want to uninstall GroundWork Distributed Monitoring Agent and all of its*

- modules?*, answer Yes to uninstall and proceed.
- Using the File Manager remove the `groundwork` directory all all sub-directories.
 - On the GroundWork server delete the GDMA client host within the *Auto Registration* host group and run a *Commit* operation.