

Tech Tip 9 - Send Downtimes Programmatically

Tech Tip 9 (4/2019) - Submit downtime via Rest API/command line



This Tech Tip applies to GroundWork Monitor version 7.2.1 which has full API support, prior versions of GroundWork Monitor used a different method.

Overview

In groundwork, downtime submitted only to the Nagios command pipe does not appear in the general Downtime user interface. This page reviews how to submit a host and service downtimes into GroundWork Monitor via the REST API, using the Linux command line, and have it appear in the Configuration > Downtimes screens to validate and audit. This solves the problem of adding all legacy downtimes from an older installation to a new one, and allows you to automate adding downtimes from other tools, or other groups in your organization who don't otherwise need access to GroundWork.

We decided to make this functionality available via a shell script, so it should be pretty clear what's going on - just look at the script. It assembles a downtime request and sends it to the API, properly formatted. Suggestions for improvements are of course welcome.

Name	Size	Creator	Creation Date	Comment
 dt_script.sh	2 kB	Thomas Stocking	Apr 19, 2019 13:39	MD5: a917d3ce7fcee97eed25354f24d8c4

Submit downtime from the command line

This script allows you to submit downtime to a GroundWork 7.2.x server from the command line.

1. Download and copy the attached script to the nagios home dir on your GroundWork server.



This can also be run from other Linux systems that have network access to the GroundWork server.

2. Make executable by nagios: (for example)

```
chmod +x dt_script.sh
```

- Run like this:

```
./dt_script.sh -S your-gw-server -u username -p password -h host-to-put-in-downtime| |null  
-s service-to-put-in-downtime| |null -g hostgroup-to-put-in-downtime| |null -r  
servicegroup-to-put-in-downtime| |null -a author -c "Description text in quotes" -D <Start  
date, e.g. "2019-04-12 21:30:00"> -E <End date, e.g. "2019-04-12 23:30:00"> -d duration
```

- You will need to feed in good input for it to work.
- The -S, -u, -p, -h, -s, -g, -r, -a, -c args are all required. Use *null* for any parameters you don't need (e.g. -g null when specifying downtime for a host).
- Start, End, Duration
 - If you don't give it start and end and duration, it will default to now for an hour.
 - If you give it a start, it will run for an hour from start.
 - And if you give it a start and duration, that will work too.
 - Start times **must** be at least 1 full minute in the future.

This will work for host (and all services) if you specify the host and leave service at null, or for a single service if you specify the host and service.

This script will also work fine for hostgroup and servicegroup downtimes.

- For a hostgroup named "foo", for example, do something like this:

```
./dt_script.sh -S your-gw-server -u username -p password -h "*" -s "*" -g foo -r null -a author  
-c "Downtime for hostgroup foo" -D "2019-04-15 21:30:00" -E "2019-04-15 23:30:00" -d 120
```

- And for servicegroup barr:

```
./dt_script.sh -S your-gw-server -u username -p password -h "*" -s "*" -g null -r barr -a author  
-c "Downtime for hostgroup foo" -D "2019-04-15 21:30:00" -E "2019-04-15 23:30:00" -d 120
```



The "*" and the null values in the appropriate positions.