

12 Tracing

The trace statement controls options and settings for the tracing function. Tracing allows a network administrator to record and view network information from a console. The console session can be saved as file for assessing network performance. GateD's tracing options may be configured anywhere in the configuration file, and include control, global, and protocol-specific tracing options.

Tracing events fall into two main categories:

- **Global.** Events that impact the GateD network as a whole. GateD uses two types of global options: those that affect only global operations and those that have potential significance to protocols.
- **Packet.** Events that record packet activity for a protocol.

When a trace option is set in the configuration file, all subsequent lower levels in the configuration file inherit the option, unless a statement is specified at the lower level. For example, a trace option set in the global statement of the configuration file would be inherited by the RIP protocol statement, unless a separate statement was added to the RIP protocol statement.

Available tracing options within the trace statement vary slightly depending on what protocol is in use. In some cases, a specific tracing statement would not make sense for a protocol. As an example, RIP does not have a state machine, so specifying the state trace option in the RIP protocol would not produce any information.

When protocols inherit their tracing options from the global tracing options, tracing levels that do not make sense (such as **parse**, **adv**, and **packet** tracing options) are masked out.

Modifying the `mpm.cmd` File

If you want to enable trace options, you will need to add a line to the `mpm.cmd` file. To enable trace options, add the following line to the `mpm.cmd` file (before the `cmlnlt` line):

```
use_gated=2
```

There are two other settings for this command. A **1** indicates GateD should run without enabling tracing. (This setting is the same as if there was no line in the `mpm.cmd` file.) A **0** indicates that GateD should not run on this switch, even if the `gated.img` and `gated.conf` are detected.

For information on how to add lines to the `mpm.cmd` file, see the chapter “Managing Files” in your switch manual.

The Trace Statement

The following is the complete trace statement, which is added to the **gated.conf** file. The trace statement is the only configuration statement that can be added anywhere in the configuration file. Each of the commands is described below.

traceoptions [*control_options*] *trace_options* [**except** *trace_options*];

This sequence of options is used to specify the name of the trace file (*trace_file*) or files and parameters about these files. Trace files can be specified as a global parameter for all of GateD, for a protocol instance, for a peer, or peers within a protocol.

The syntax conventions used in the above statement are fully described in Chapter 3 of this manual.

control_options

Specifies options that control the appearance of trace events on the console screen. Valid values are:

nostamp	Specifies that a timestamp should not be prepended to all trace lines.
----------------	--

trace_options [**except** *trace_options*]

This command is used to enable a broad class of tracing and then disable more specific options. There are several types of tracing options available for this command. Listed below are the global tracing options (including global significant and protocol significant) and packet tracing options. For details of the available options for a specific protocol, see the chapter in this manual that covers the protocol.

Global tracing options are options affecting only global operations and options that have potential significance to protocols. The trace flags that have only global significance are:

parse	This modifier traces GateD's lexical analyzer and parser. This is mostly used by GateD developers for debugging.
adv	This modifier traces the allocation and freeing of policy blocks. This is mostly used by GateD developers for debugging.
none	Specifies that all tracing should be shut down for this protocol or peer.

The trace flags that have potential protocol significance are:

all	This modifier is used to specify all of the following modifiers
general	This modifier is used to specify both the normal and route modifiers.
state	This modifier traces state machine transitions in the protocols. This is not valid for RIP.
normal	This modifier traces normal protocol occurrences. Abnormal protocol occurrences are always traced.
policy	This modifier traces the application of protocol and user specified policy to routes being imported and exported.
task	This modifier traces system interface and processing associated with this protocol or peer.
timer	This modifier traces timer usage by this protocol or peer.
route	This modifier traces routing table changes for routes installed by this protocol or peer.

The trace flags that have packet significance are:

detail	This modifier must be specified before send or recv (see below). Normally packets are traced in a terse form of one or two lines. When detail is specified, a more verbose format provides further detail on the contents of the packet.
send or recv	These options limit the tracing to packets that are either sent (send) or received (recv). Without these options, both sent and received packets will be traced.

Trace Statement Example

The following is an example of a trace statement:

```
traceoptions nostamp all except policy task ;
```

In the above statement:

- The **nostamp** token signifies that no timestamp is added to the trace logs.
- The **all** variable coupled with the **except policy task** modifiers states that all variables possible should be logged except **policy** and **task**.

