



## Chapter 2 Trace Statements and Global Options

### traceoptions

#### Name

**traceoptions** - specifies packet tracing options

#### Syntax

```
traceoptions [ trace_file [ replace ]  
  [ size tracesize [ k | m ] files tracefiles ] ]  
  [ nostamp ] trace_options [ except trace_options ] ;  
traceoptions none;
```

#### Parameters

*trace\_file* - specifies the file to receive tracing information. If this file name does not begin with a slash ('/'), the directory where GateD was started is prepended to the name. Specifying "**stderr**" writes trace output to standard error.

**replace** - specifies to start tracing by truncating an existing file. The default is to append to an existing file.

**size** *tracesize* [ **k** | **m** ] **files** *tracefiles* - limits the maximum size of the trace file to the specified *tracesize* (minimum 10k). When the trace file reaches the specified size, the file is renamed to file.0, then file.1, then file.2, up to the maximum number of files as specified by *tracefiles*. (The minimum specification is 2.)

**k** - specifies the file size in kilobytes

**m** - specifies the file size in megabytes

**files** *tracefiles* - specifies the maximum number of files

**nostamp** - specifies that a timestamp should not be prepended to all trace lines

*trace\_options* - described below

The *trace\_options* that have only global significance are:

**parse** - specifies to trace the lexical analyzer and parser. **parse** is used mostly by GateD developers for debugging configuration parser processing.

**adv** - specifies to trace the allocation of and freeing of policy blocks. **adv** is used mostly by GateD developers for debugging the use of **adv-entry** structures in parsing.

**symbols** - specifies to trace symbols read from the kernel at startup. The only useful way to specify this level of tracing is via the **-t** option on the command line, because the symbols are read from the kernel before parsing the configuration file.

**iflist** - specifies to trace the reading of the kernel interface list, to trace the initial read of the interface information. Specify **iflist** with the **-t** option on the command line, because the first interface scan is done before reading the configuration file.

The *trace\_options* that have potential significance to protocols are:

**none** - specifies that all tracing should be turned off for this protocol or peer

**all** - specifies to turn on **detail**, **packets**, and all of the following:

**general** - specifies to trace both **normal** and **route**

**normal** - specifies to trace normal protocol occurrences. Abnormal protocol occurrences are always traced.

**route** - specifies to trace routing table changes for routes installed by this protocol or peer

**state** - specifies to trace state machine transitions in the protocols

**policy** - specifies to trace application of protocol and user-specified policy to routes being imported and exported

**task** - specifies to trace system interface and processing associated with this protocol or peer

**timer** - specifies to trace timer usage by this protocol or peer

Tracing of packets is very flexible. For any given protocol, there are one or more options for tracing packets. All protocols allow use of the **packets** keyword for tracing all packets sent and received by the protocol. Most protocols have other options for limiting tracing to a useful subset of packet types. These tracing options can be further controlled with the following modifiers:

**detail** - must be specified before **send** or **receive**. 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 **receive** - limit the tracing to packets sent or received. If neither is specified, both sent and received packets will be traced.

**except trace\_options** - used to enable a broad class of tracing and then disable more specific options

## Description

Trace statements control tracing options. GateD's tracing options can be configured at many levels. Tracing options include the file specifications and global and protocol-specific tracing options. Unless overridden, tracing options from the next higher level are inherited by lower levels. For example, BGP peer tracing options are inherited from BGP group tracing options, which are inherited from global BGP tracing options, which are inherited from global GateD tracing options. At each level, additional tracing specifications override the inherited options.

Not all of the above *trace\_options* apply to all of the protocols. In some cases, their use would not make sense (for instance, RIP does not have a state machine) and, in some cases, the requested tracing has not been implemented. Currently, you cannot specify packet tracing from the command line because a global option for packet tracing could create too much output.

When protocols inherit their tracing options from the global tracing options, tracing options that do not make sense (such as `parse`, `adv` and `packet` tracing options) are masked out. Global tracing statements have an immediate effect, especially parsing options that affect the parsing of the configuration file. Tracing values inherited by protocols specified in the configuration file are initially inherited from the global options that are currently in effect as the protocol configuration entries are parsed, unless they are overridden by more specific options.

After the configuration file is read, protocol tracing options that were not explicitly specified are inherited from the global options in effect at the end of the configuration file. When more than one `traceoptions` line is used in a section, the "last" `traceoptions` line to be parsed by GateD is the one that takes effect. In the case of global tracing, any trace files specified in any `traceoptions` line will be created, but tracing will cease for that file when the next `traceoptions` line is parsed.

The `size` and `file` options in subsequent `traceoptions` statements with the same file name modify the previously set values for the files. If a subsequent `traceoptions` statement refers to the same file but does not specify `size` and `file` values, the size limit for the file is set to unlimited. Setting `size` to 0 also sets the size limit to unlimited. In this latter case, the `files` keyword and value can be omitted.

## Defaults

- Tracing is written to standard error.
- There is no limit on the file size.
- `nostamp` is disabled.
- There are no trace options.

## Context

global

## Examples

### Example 1

The first example traces everything but policy to "icmp/GateD.log" (iflist is superfluous).

```
traceoptions "/tmp/GateD.log" replace size 100 K files 3
iflist all except policy;
```

### Example 2

```
traceoptions "log" size 1024k files 2;
# Once the log file reaches 1 megabyte in length, it is
# moved to log.0, and the log is recreated.
#
```

```
# Again, when the log file file reaches 1 megabyte in length,  
# log.0 is renamed log.1, log is renamed log.0,  
# and the log is recreated.  
#  
# Again, when the log file reaches 1 megabyte in length,  
# log.1 is deleted, log.0 is renamed log.1, log is renamed log.0,  
# and the log is recreated.
```

### Example 3

```
traceoptions "/tmp/GateD.log"  
# The log file is /tmp/GateD.log. It will grow in size until  
# space on /tmp is depleted
```

### Example 4

```
traceoptions policy;  
traceoptions all except general;
```

### See Also

- rip** on page 49
- bgp** on page 227
- ospf** on page 95
- isis** on page 153
- icmp** on page 313
- redirect** on page 319
- kernel** on page 327
- dvmrp** on page 361
- pim** on page 379
- mpbgp** on page 435
- igmp** on page 527
- smux** on page 715