Configuration files

an in-depth guide on tBB configuration files

Introduction.

If you wish to configure tBB settings, you have to use the configuration files tBB is instructed to use. These files are always located in the root tBB folder. Please check where this folder is located before continuing reading.

Please, note that you're free to change settings in these files at any time, but if you want your changes to take effect you'll to have to restart tBB.

Configuration files types.

There are two different kinds of configuration files: the default configuration file and optional network-specific configuration files. The default configuration file comes along with tBB and configures it to use a general-purpose-suitable configuration. You may want to check if the built-in default configuration suits your needs and preferences before start using it.

The network-specific configuration files are only invoked when tBB had been launched passing a network as an argument. In this case, tBB is instructed to look for a specific configuration file in the main tBB folder. If no such file is found, tBB will fall-back to the default configuration file.

Any field that is not set in the network-specific configuration file will fall-back to the value set in the default configuration file.

Network-specific configuration files naming conventions.

Specific configuration files naming conventions follow the scans naming conventions. For instance, if you want to create a configuration file for network 192.168.100.0/24 you're going to need to create a file named “`config_192.168.100.0\24-256.json`”. Please note the backslash replacing the forward slash (forward slash is invalid for the Unix file name conventions). Also note the given network length in the filename.

This rigid naming conventions allow tBB to use the correct configuration file for every network you may want to monitor.

Configuration fields.

The configuration files are in JSON format and therefore nested as of the nature of JSON (see RFC#7159).

You may use the following macros while defining a field value: “`{default_time_format}`”, “`{frontends_socket_port}`”. They will be replaced by the appropriate value at runtime.

Italic example values are the default values.

These are the fields tBB accepts, divided in the appropriate nested sections, given in no particular order:

`Root-level`:

Field name	*Description*	*Example values*

`networkIp`	The network tBB is called to monitor.	`192.168.100.0/24`
		`192.168.100.0/24-10`

least_record_update_seconds	Maximum amount of time for which tBB will not re-perform a complete scan on startup (in seconds).	10 → 10 seconds *3600 → 1 hour* 86400 → 1 day
frontends_socket	Front-ends communication section. See below.	…
logging	Logging section. See below.	…
tracker	Tracker section. See below.	…
serialization	Serialization section. See below.	…

frontends_socket:

Field name	*Description*	*Example values*
host	IP from which open the socket.	192.168.100.101 *localhost*
port	Port number from which open the socket.	*1984* 65000
maximum_port_lookup	Maximum number of times tBB will look for another open port if the specified one isn't available.	*20* 0 100
ssl	Enable/disable SSL communication.	*true* false
do_checks	Enable/disable host name checking with SSL enabled. If enabled, certificates must be valid.	*true* false

logging:

Field name	*Description*	*Example values*
version	Always set this field to *1*.	*1*
disable_existing_loggers	Always set this field to *false*.	*false* true
formatters	Formatters section. See below.	…
handlers	Handlers section. See below.	…
loggers	Loggers section. See below.	…

formatters:

Note: this field is a list. Each item configures one logging formatter that can be identified by the name given to the field. Shown are the item's fields.

Field name	*Description*	*Example values*
format	String to format logging upon. For further details see related documentation.	…
datefmt	String to format dates upon (optional). Macro "{default_time_format}" is available. For further details see related documentation.	*{default_time_format}* …

`handlers`:

Note: this field is a list. Each item configures one logging handler that can be identified by the name given to the field. Shown are the item's fields.

Field name	*Description*	*Example values*
`level`	One of "`DEBUG`", "`INFO`", "`WARNING`", "`ERROR`", "`CRITICAL`". For further details see related documentation.	…
`class`	A valid logging handler class. For further details and a list of available handler classes, see related documentation.	…
`formatter`	One of the formatters defined in "`logging`" → "`formatters`"	…
…more…	Other fields can possibly be defined for each handler, but they are strictly related to the class you're using. For further details on class-dependent fields, please see related documentation.	…

`loggers`:

Note: this field is a list. Each item configures one logger that can be identified by the name given to the field. The name "" (blank) and the name "`root`" identify the default logger. Shown are the item's fields.

Field name	*Description*	*Example values*
`handlers`	List of handlers to use for this logger as defined in "`logging`" → "`handlers`".	*["console", "file", "syslog"]*
`level`	Logging level to use for this logger. One of "`DEBUG`", "`INFO`", "`WARNING`", "`ERROR`", "`CRITICAL`". For further details see related documentation.	*INFO* `WARNING`
`propagate`	Enable/disable logging propagation. For further details see related documentation.	*true* `false`

`serialization`:

Field name	*Description*	*Example values*

`indent`	Indent value for pretty saving of scan files.
	If set to `null`, it will also prefent `\n`'s from being written to file.
		4
		`null`
		`2`

`do_sort`	Enable/disable item sorting within scan files.	*true* `false`

`tracker`:

Field name	*Description*	*Example values*
`hosts`	Number of hosts sub-networks must be	*16* `64`

	divided into. Must be a valid network length.	
`enable_notifiers`	Enable/disable notifiers.	*`true`* `false`
`auto_ignore_broadcasts`	Enable/disable automatic broadcasts ignore. If enabled, when a broadcast is detected during a scan, it will be ignored in the next ones.	*`true`* `false`

`time_between_checks`	Divided into “`minutes`” and “`seconds`”.	*`{“minutes”: 0, “seconds”: 2}`*
		`{“minutes”: 1, “seconds”: 30}`

`maximum_seconds_randomly_added`	Maximum amount of time to add randomly to “`time_between_checks`” (in seconds). Must be a positive integer.	*`2`*
		`10`

`ignore`	List of IPs to ignore.	*`[]`*
`ignore_mac`	List of MACs to ignore.	*`[]`*
`arp`	ARP section. See below.	…
`disoveries`	Discoveries list. See below	…

`arp`:

Field name	*Description*	*Example values*
`count`	Number of ARP broadcasts to emit.	*`3`*

`timeout`	Maximum amount of time in which to wait for a response (in seconds). Must be a positive integer.	*`2`*
		`4`

`quit_on_first`	Stop listening for responses at first response.	*`true`* `false`

`discoveries`:

Note: this field is a list. Each item configures one discovery method. Shown are the item's fields.

Field name	*Description*	*Example values*
`type`	One of “`icmp`”, “`syn`”.	…

`count`	Number of requests to send.
	If “`flood`” is enabled, it represents the number of responses to receive before returning.
	Only available for type “`icmp`”.
		`1`
		`4`

`timeout`	Maximum amount of time to wait for a response (in seconds). Must be a positive integer.
	A higher value in this field represent a more reliable check, but also a slower one.
		`4`
		`1`
		`10`

`flood`	Enable/disable flood ping mode. Only available for type “`icmp`”.	*`true`* `false`

`ports`	Ports to check. Must be of string type. Only available for type “`syn`”.	*“22”* “80”
`enabled`	Enable/disable discovery method.	*true* false