Download and Setup Guide

Tom Kashefi

March 24, 2025

Download and Setup Guide

By Tom Kashefi

Download the latest version from the releases page.

1. Add the argument `-javaagent:/absolute/path/to/ja-netfilter.jar` (replace with your actual file path).

   - Add it as an argument to the Java command, e.g.:

     ```
     java -javaagent:/absolute/path/to/ja-netfilter.jar -jar executable_jar_file.jar
     ```

   - Some applications support a JVM Options file; you can add it as a line in that file.
   - **Warning**: Do not include unnecessary whitespace characters!

2. Alternatively, run `java -jar /path/to/ja-netfilter.jar` to use attach mode.

3. For Java 17, include at least these JVM options:

   ```
   --add-opens=java.base/jdk.internal.org.objectweb.asm=ALL-UNNAMED
   --add-opens=java.base/jdk.internal.org.objectweb.asm.tree=ALL-UNNAMED
   ```

4. Edit your plugin configuration files:

   - Locate the `${lower plugin name}.conf` file in the `config` directory where `ja-netfilter.jar` resides.
   - The `config`, `logs`, and `plugins` directories can be customized via `javaagent` arguments, e.g.:

     ```
     -javaagent:/path/to/ja-netfilter.jar=appName
     ```

     This sets the directories to `config-appname`, `logs-appname`, and `plugins-appname`.
   - If no arguments are provided, they default to `config`, `logs`, and `plugins`.
   - This approach keeps your `config`, `logs`, and `plugins` directories lean and organized.

5. Run your Java application and enjoy!

ja-netfilter 2022.2.0

A Javaagent Framework by Tom Kashefi

Usage

Config File Format [ABC]

```
# Specify the section name, e.g.:
[URL]
EQUAL,https://someurl

[DNS]
EQUAL,somedomain
```

Comparison options:

- `EQUAL`: Use `equals` to compare

- `EQUAL_IC`: Use `equals`, ignore case

- `KEYWORD`: Use `contains` to compare

- `KEYWORD_IC`: Use `contains`, ignore case

- `PREFIX`: Use `startsWith` to compare

- `PREFIX_IC`: Use `startsWith`, ignore case

- `SUFFIX`: Use `endsWith` to compare

- `SUFFIX_IC`: Use `endsWith`, ignore case

- `REGEXP`: Use regular expressions to match

Debugging

- By default, ja-netfilter does not output debugging information.

- To enable it:
    - Add the environment variable `JANF_DEBUG=1` (log level) before starting, or
    - Add the system property `-Djanf.debug=1` (log level).

- Log levels: `NONE=0`, `DEBUG=1`, `INFO=2`, `WARN=3`, `ERROR=4`.

- Debugging output goes to the console by default.

- To change the output medium:
    - Add the environment variable `JANF_OUTPUT=value` before starting, or
    - Add the system property `-Djanf.output=value`.

- Output medium values:
    - `NONE=0`, `CONSOLE=1`, `FILE=2`, `CONSOLE+FILE=3`, `WITH_PID=4`
    - Example: Console + file + PID file name = 1 + 2 + 4 = 7, so use `-Djanf.output=7`.

For Developers

- Explore the scaffold project designed for the plugin system.

- Compile your plugin, publish it, and let your imagination run wild!

For Users

- Download the plugin's `.jar` file.

- Place it in the `plugins` subdirectory where `ja-netfilter.jar` is located.

- Enjoy the new capabilities introduced by the plugin.

- Note: If the file suffix is `.disabled.jar`, the plugin will be disabled.

Debug Info

Debug output and plugin system details are available when debugging is enabled.