

# Excluding classes as an agent parameter

## Information:

### Environment

This KB article relates to the java agent only.  
The collector debug flag to exclude classes is available since release 6.2

### Symptoms

The point of excluding a class already at the agent level is to prevent it from getting sent to the collector in the first place (instead of just preventing it to be instrumented there with a sensor exclude rule).

This is mainly useful to keep the class cache size small by excluding dynamically generated classes.

### Solution

## Excluding classes as an agent parameter

The exclusion rules are passed to the agent either by an agentpath parameter ("exclude=") e.g.:

```
agentpath:/path/to/agent.so=name=AgentName,server=server-host,exclude=contains:ClassToExclude,loglevel=FINE
```

or as a [collector debug option](#) ("com.dynatrace.diagnostics.agent.exclude").

If you modify the agentpath string, the application needs to be restarted after setting this parameter. The collector debug flag is effective immediately, but already instrumented classes will stay instrumented until after the next restart.

## Syntax:

The grammar for the exclusion agentpath parameter is as follows:

```
agentpath-parameter      =  "exclude=", exclude-rule-list
exclude-rule-list       =  exclude-rule, {";", exclude-rule}
exclude-rule             =  operator, ":", string-for-comparison
operator                 =  "contains" | "starts" | "ends" ;
string-for-comparison    =  "... " {"/", "..."} ;
```

For the debug flag, use only the "exclude-rule-list" as content of the flag.

Noteworthy in this grammar are:

- package parts are separated by "/", not ".", like in com/dynatrace/diagnostics/server/
- exclude rules are separated by ";" characters
- I've left out the optional quotation marks around the exclude-rule-list. Please don't use them, they shouldn't be necessary and might cause problems.

## Examples:

**If you want to exclude all classes containing the string "CGLIB", the parameter should look like this:**

```
agentpath:/path/to/agent.so=name=AgentName,server=server-host,exclude=contains:CGLIB,loglevel=FINE
```

## If you want to exclude all classes containing \$Proxy:

```
exclude=contains:$Proxy
```

Please note that the "\$" sign has a special meaning in some shell interpreters and should be escaped accordingly if that's an issue in your case (you'll get an error message on app start up)

## If you want to exclude both cases:

```
exclude=contains:CGLIB;contains:$Proxy
```

Please note that the logical combination here is always a simple "or", that is a class will be excluded if it either contains the string "CGLIB" or the string "\$Proxy"

## An example with a package name:

```
exclude=starts:com/dynatrace/diagnostics/server/;ends:MyClass
```

Also here, we'll exclude all classes that start with `com.dynatrace.diagnostics.server` **or** end with `MyClass`, not as one could think the combination of the two conditions.

## Exclude-rule on the collector (AppMon 6.2+ only)

Since AppMon 6.2, agent-side exclude rule can also be specified with a debug flag on the collector. The rule uses the same syntax as the agent-side exclude rule specified on the agent command line.

```
com.dynatrace.diagnostics.agent.exclude=<exclude-rule-list>
```

## How to set a debug flag on a collector: [Setting a AppMon Collector Debug Flag](#)

### Example (in dtcollector.ini):

```
-Dcom.dynatrace.diagnostics.agent.exclude=contains:ClassNameToBeExcluded
```