

Monitoring: idea

The core idea of monitoring is an ability to monitor some *event* and perform corresponding *action* if that event occurred.

Configuration

Monitoring can be configured in the Yaml configuration file, in the section `mapMonitors`:

```
mapMonitors:
  chain.creation.block:
    monitor: ...
    actions:
      - ...
      - ...
```

In this example "chain.creation.block" is a logger name, and we will monitor messages sent to this logger.

Event

Monitoring event is defined in the configuration as a logical expression, for example:

```
monitor: (monitMe >= (42))
```

So if the value of `monitMe` will reach 42 - event occurs.

It's possible to use the following compare operators: `<=`, `<`, `>=`, `>`, `=`, `!=`, `/=`.

Composition and arithmetic

Comparisons can be composed using `And`, `Or` and `Not`, for example:

```
monitor:
  ((monitMe < (2)) Or (monitMe >= (4)))
```

It's possible to do basic arithmetic in comparisons using operators `+`, `-` and `*`, for example:

```
monitor: (monitMe > ((19) * (13)))
```

Measurable

Any **Measurable** value can be used in comparisons, for example:

```
monitor: (time > (22 s))
```

or

```
monitor: (f.size > (22 MB))
```

Here `s` corresponds to Seconds and `MB` to megabytes.

Action

When monitoring event occurred, corresponding action is performing.

Action is defined in the configuration, for example:

```
actions:
  - CreateMessage Warning "Alert msg!"
```

The following actions are supported:

1. `CreateMessage` — creates a message with defined log severity.
2. `SetGlobalMinimalSeverity` — sets global minimal log severity.
3. `AlterSeverity` — alters log severity for particular logger.

Precondition

It's possible to define precondition expression for monitoring. This precondition is optional, but if it's defined — monitoring will be performed only if precondition returns `True`. For example:

```
monitor-if: (monitMe.fcount >= (10))
monitor: (monitMe >= (34))
```

In this case the first 10 `monitMe` values will be skipped for monitor analysis.

Please note that technically precondition expression is the same one as an event expression, and can contain the same compare operators, logical compositions and arithmetic operators.