NAME

Monitoring::Availability − Calculate Availability Data from Nagios / Icinga and Shinken Logfiles.

SYNOPSIS

```
use Monitoring::Availability;
my $ma = Monitoring::Availability->new();
```

DESCRIPTION

This module calculates the availability for hosts/server from given logfiles. The Logfileformat is Nagios/Icinga only.

REPOSITORY

```
Git: http://github.com/sni/Monitoring-Availability
```

CONSTRUCTOR

new ([ARGS])

Creates an `Monitoring::Availability` object. `new` takes at least the logs parameter. Arguments are in key-value pairs.

rpttimeperiod
: report timeperiod. defines a timeperiod for this report. Will use 24x7 if not specified.

assumeinitialstates
: Assume the initial host/service state if none is found, default: yes

assumestateretention
: Assume state retention, default: yes

assumestatesduringnotrunning
: Assume state during times when the monitoring process is not running, default: yes

includesoftstates
: Include soft states in the calculation. Only hard states are used otherwise, default: no

initialassumedhoststate
: Assumed host state if none is found, default: unspecified

  valid options are: unspecified, current, up, down and unreachable

initialassumedservicestate
: Assumed service state if none is found, default: unspecified

  valid options are: unspecified, current, ok, warning, unknown and critical

backtrack
: Go back this amount of days to find initial states, default: 4

showscheduleddowntime
: Include downtimes in calculation, default: yes

timeformat
: Time format for the log output, default: `%s`

verbose
: verbose mode

breakdown
: Breakdown availability into 'months', 'weeks', 'days', 'none'

  adds additional 'breakdown' hash to each result with broken down results

METHODS

calculate

```
calculate()
```

Calculate the availability

start
: Timestamp of start

end Timestamp of end

log_string
String containing the logs

log_file
File containing the logs

log_dir
Directory containing *.log files

log_livestatus
Array with logs from a livestatus query

```
a sample query could be:
selectall_arrayref(GET logs...\nColumns: time type options, {Slice => 1})
```

log_iterator
```
Iterator object for logentry objects. For example a L<MongoDB::Cursor> object
```

hosts
array with hostnames for which the report should be generated

services
array with hashes of services for which the report should be generated. The array should look like this:

```
[{host => 'hostname', service => 'description'}, ...]
```

initial_states
if you use the ‘‘current’’ option for initialassumedservicestate or initialassumedhoststate you have to provide the current states with a hash like this:

```
{
  hosts => {
   'hostname' => 'ok',
   ...
  },
  services => {
   'hostname' => {
       'description' =>  'warning',
       ...
    }
  }
}
```

valid values for hosts are: up, down and unreachable

valid values for services are: ok, warning, unknown and critical

get_condensed_logs
```
get_condensed_logs()
```

returns an array of hashes with the condensed log used for this report

get_full_logs
```
get_full_logs()
```

returns an array of hashes with the full log used for this report

BUGS

Please report any bugs or feature requests to <http://github.com/sni/Monitoring−Availability/issues>.

DEBUGING

You may enable the debug mode by setting MONITORING_AVAILABILITY_DEBUG environment variable. This will create a logfile: /tmp/Monitoring−Availability−Debug.log which gets overwritten with every calculation. You will need the Log4Perl module to create this logfile.

SEE ALSO

You can also look for information at:

- Search CPAN

  <http://search.cpan.org/dist/Monitoring−Availability/>
- Github

  <http://github.com/sni/Monitoring−Availability>

AUTHOR

Sven Nierlein, <nierlein AT cpan DOT org>

COPYRIGHT AND LICENSE