

Configure logging for OpenAthens SP on Apache and .NET

Log configuration files

Logging is configured via an XML-based log configuration file:

Apache: /etc/httpd/conf.d/oaLogSP.xml

.NET: C:\Program Files (x86)\Eduserv\OpenAthens.Net\conf\oaLogSP.xml (typical)

For Java see: [Configure logging in OpenAthens SP on Java on Linux](#) or [Configure logging in OpenAthens SP on Java on Windows](#)

These examples use the Apache version

Example config

```
<?xml version="1.0" ?>
<atacamaLogConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="urn:mace:eduserv.org.uk:aim:schemas:atacama-log-config:1.1"
  xmlns:sk="urn:mace:eduserv.org.uk:aim:schemas:atacama-log-config:1.1:ext:unixsocket">

  <define name="directory" value="/var/log/openathens" />

  <logger name="platform" xsi:type="loggerType">
    <level>INFO</level>
    <output xsi:type="fileOutputter">
      <directory>${directory}</directory>
      <rollover>DAILY</rollover>
    </output>
  </logger>

  <logger name="stderr" xsi:type="loggerType">
    <level>WARN</level>
    <output xsi:type="fdOutputter">
      <to>STDERR</to>
    </output>
  </logger>

  <logger name="login" channels="login" xsi:type="loggerType">
    <level>AUDIT</level>
    <level>ERROR</level>
    <output xsi:type="fileOutputter">
      <directory>${directory}</directory>
      <rollover>DAILY</rollover>
    </output>
  </logger>

  <!-- Uncomment to enable logging of protocol-level messages
  <logger name="protocol" channels="protocol" xsi:type="loggerType">
    <level>INFO</level>
    <output xsi:type="fileOutputter">
      <directory>${directory}</directory>
      <rollover>DAILY</rollover>
    </output>
  </logger>
  -->

</atacamaLogConfig>
```

Loggers

A log configuration may specify multiple loggers. A logger specifies a level and an outputter type. Loggers must be uniquely named.

Log levels

As is normal with log frameworks, log events emitted by the Atacama platform are classified at a particular level of severity. The available log levels, in ascending severity are:

- DEBUG
- INFO
- WARN
- ERROR
- FATAL

There is an `AUDIT` level, but this must be specified individually as it does not form part of the above hierarchy.

Level inclusion conditions

A logger will emit log events matching one or more levels. By default, specifying a level will include events at that level or levels that are more severe. For instance, a level of `WARN` includes `WARN`, `ERROR` and `FATAL` events. It is also possible to specify a precise list of levels by specifying an exact inclusion condition via the `inc` parameter. For example, a list of the following two level elements is used to emit log events matching `ERROR` and `INFO` events only:

```
<level inc="exact">ERROR</level>
<level inc="exact">INFO</level>
```

Outputters

Each logger must have an outputter associated with it. As the name suggests, an outputter is used to output the messages to a particular destination such as the system console or a file.

The available outputters are:

File outputter

Outputs to a file on the local file system. This file may rollover on a `HOURLY`, `DAILY`, `WEEKLY` or `MONTHLY` basis and is given a unique name based on the logger name, and date. The directory where the log files will be written must also be specified.

```
<output xsi:type="fileOutputter">
  <directory>/var/log/openathens</directory>
  <rollover>MONTHLY</rollover>
</output>
```

Pipe outputter

Pipes log messages to an external program. This could be used in conjunction with other programs such as `Cronolog` or `.`

```
<output xsi:type="pipeOutputter">
  <command>cronolog /var/log/openathens/%Y-%m-openathens_log</command>
</output>
```

File descriptor outputter

Sends logs to `stderr` or `stdout`.

```
<output xsi:type="fdOutputter">
  <to>STDOUT</to>
</output>
```

Windows event logging

To do this you must firstly declare an additional XML namespace in the log configuration file. This namespace is used to specify Windows-specific extensions. Then use the `eventLog` outputter declared in this namespace. It is optionally possible to specify the name of a separate server if events are to be sent to a server other than the local host.

```
<atacamaLogConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:wx="urn:mace:eduserv.org.uk:aim:schemas:atacama-log-config:1.0:ext:windows"
xmlns="urn:mace:eduserv.org.uk:aim:schemas:atacama-log-config:1.0">
  <logger name="logger2" xsi:type="loggerType">
    <level>INFO</level>
    <output xsi:type="wx:eventLog"/>
  </logger>
</atacamaLogConfig>
```