

Configuring the Log and Audit Trail

- [Default log file location](#)
- [Changing the log file storage location](#)
- [Changing the log file roll over strategy](#)
- [Enabling log output to *.html files](#)
- [Enabling and configuring the Audit Trail mode](#)
- [Enabling DEBUG log mode for Active Directory/SAML authentication issues](#)
- [Enabling DEBUG and SQL trace logs](#)
 - [1.x](#)
 - [2.x](#)



Summary: this page explains where the dashboard/RestAPI log files are located, how to change their storage location if required and how to enable the Audit Trail facility.

Default log file location

By default the dashboard/RestAPI will place all relevant log files (i.e. logs relative to the functioning of the dashboard) in the following location:

```
WAR 1.x
CATALINA_HOME\webapps\<<dashboard>\logs

WAR 2.x
CATALINA_HOME\webapps\<<dashboard>\WEB-INF\classes\logs

ZIP 2.x
<unpacked_zip>\configurations\logs
```

Logs are provided in two different formats:

Log files	Description	Notes
restapi.log restapi.html	Records all web service interaction (errors, warnings etc.)	Note that the *.html log files will not be used in the default "out of the box" configuration. If you would like to have log messages sent to the *.html files as well as the *.log files, please see Enabling log output to *.html files .
restapi.perf.log restapi.perf.html	Records performance information about user interaction. Not generated in CAST Dashboard 2.x.	
restapi.audit.log	Records successful user logins, failed logins, logouts, web application startup/stop. This log is not active by default.	See Enabling and configuring the Audit Trail mode section below for more information about this.

Changing the log file storage location

You do not need to change the location of the log files unless you specifically need to use a different location other than the default. To do so, you need to modify the following file with a text editor:

```
WAR 1.x
CATALINA_HOME\webapps\<dashboard>\WEB-INF\log4j2.xml

WAR 2.x
CATALINA_HOME\webapps\<dashboard>\WEB-INF\classes\log4j2-spring.xml

ZIP 2.x
<unpacked_zip>\configurations\log4j2-spring.xml
```

To change the location of the log folder search for the following line in the file:

```
WAR 1.x - default location is within the web application parent folder
<Property name="logPath">${web:rootDir}/logs</Property> >>>

WAR and ZIP 2.x - default location is a folder at the same level as the log4j2-spring.xml file
<Property name="logPath">${log4j:configParentLocation}/logs</Property>
```

Change **/logs** to the required location (for example **/test**) within the web application file hierarchy:

```
WAR 1.x
<Property name="logPath">${web:rootDir}/test</Property> >>>

WAR and ZIP 2.x
<Property name="logPath">${log4j:configParentLocation}/test</Property>
```

Following any changes you make, **save the .xml file** and then **restart** your application server so that the changes are taken into account. Your log files will now be stored in the new location. For example:

```
WAR 1.x
CATALINA_HOME\webapps\<dashboard>\test

WAR 2.x
CATALINA_HOME\webapps\<dashboard>\WEB-INF\classes\test

ZIP 2.x
<unpacked_zip>\configurations\test
```

Changing the log file roll over strategy

All log files that are generated have a default roll over strategy as follows:

- When the initial log file (as listed in the **Default log file location** section above) reaches 10MB in size, logging will cease to this file and a new dated log file (with the file name in the format **restapi-{yyyy-MM-dd}.log**) will be created and logging will continue in the new file.
- **5 previous log files** will be retained in the default log file location. Any subsequent roll overs will cause the oldest log file to be deleted, keeping the number of log files retained to 5.

This strategy can be modified as follows modify the following file with a text editor:

```
WAR 1.x
CATALINA_HOME\webapps\<dashboard>\WEB-INF\log4j2.xml

WAR 2.x
CATALINA_HOME\webapps\<dashboard>\WEB-INF\classes\log4j2-spring.xml

ZIP 2.x
<unpacked_zip>\configurations\log4j2-spring.xml
```

Find the following two elements which control the strategy (there are two elements per log file):

```
<SizeBasedTriggeringPolicy size="10 MB"/>
<DefaultRolloverStrategy max="5"/>
```

Modify the options as necessary. Following any changes you make, **save the .xml file** and then **restart** your application server so that the changes are taken into account.

Enabling log output to *.html files

By default the log system is configured to output log messages **only to *.log files**. If you require messages to be output to ***.html files as well**, you need to modify the following file with a text editor:

```
WAR 1.x
CATALINA_HOME\webapps\<dashboard>\WEB-INF\log4j2.xml

WAR 2.x
CATALINA_HOME\webapps\<dashboard>\WEB-INF\classes\log4j2-spring.xml

ZIP 2.x
<unpacked_zip>\configurations\log4j2-spring.xml
```

Add the line **<AppenderRef ref="htmlout"/>** to the following **<Logger>** elements in the file, this will give you the following:

```
WAR 1.x

<Logger name="com.castsoftware.adg.webservice" level="info" additivity="false">
  <AppenderRef ref="textout" />
  <AppenderRef ref="htmlout" />
</Logger>
<Logger name="com.castsoftware.adg.webservice.servlet" level="error" additivity="false">
  <AppenderRef ref="textout" />
  <AppenderRef ref="htmlout" />
</Logger>
<Logger name="org.springframework" level="info" additivity="false">
  <AppenderRef ref="textout" />
  <AppenderRef ref="htmlout" />
</Logger>
<Logger name="org.springframework.jdbc.core.JdbcTemplate" level="error" additivity="false">
  <AppenderRef ref="textout" />
  <AppenderRef ref="htmlout" />
</Logger>
<Logger name="org.springframework.jdbc.core.StatementCreatorUtils" level="error" additivity="false">
  <AppenderRef ref="textout" />
  <AppenderRef ref="htmlout" />
</Logger>

WAR and ZIP 2.x

<Logger name="com.castsoftware.adg.webservice" level="info" additivity="false">          <!-- write in restapi.
log all messages from com.castsoftware.adg.webservice with level <= INFO -->
  <AppenderRef ref="textout" />
  <AppenderRef ref="stdout" />
  <AppenderRef ref="htmlout" />
</Logger>
<Logger name="org.springframework" level="info" additivity="false">
  <AppenderRef ref="textout" />
  <AppenderRef ref="stdout" />
  <AppenderRef ref="htmlout" />
</Logger>
```



Note that there are other **<Logger>** elements in the file, but these do not need modifying.

Enabling and configuring the Audit Trail mode

By default, the various log outputs do not contain any information about **successful user logins, failed logins, logouts, web application startup/stop etc.** If you would like to see this information in a log file for security reasons, then you can enable the **Audit Trail** feature. To do so, you need to modify the following file with a text editor:

```
WAR 1.x
CATALINA_HOME\webapps\<dashboard>\WEB-INF\log4j2.xml

WAR 2.x
CATALINA_HOME\webapps\<dashboard>\WEB-INF\classes\log4j2-spring.xml

ZIP 2.x
<unpacked_zip>\configurations\log4j2-spring.xml
```

Search for the following line in the file:

```
<Property name="auditLevel">OFF</Property>
```

Change the **OFF** value to **ALL**:

```
<Property name="auditLevel">ALL</Property>
```

Following any changes you make, **save the .xml file** and then **restart** your application server so that the changes are taken into account. Messages relating to **successful user logins, failed logins, logouts, web application startup/stop** will now be displayed in the **restapi.audit.log** file in the following format:

```
[ Date | ip address | level | user behavior | rest URL | user name | user role | delay time | source |
application
```

For example:

```
2015-03-19 18:05:07,488 | 192.168.16.73 | INFO | Application startup | SLI
2015-03-19 18:05:29,602 | 192.168.16.73 | INFO | Login successful | http://tomcat/testContext/rest/login
CIO [NO_ROLE] 18 Engineering
2015-03-19 18:05:29,602 | 192.168.16.73 | INFO | Login successful | http://tomcat/testContext/rest/user CIO
[NO_ROLE] 0 Engineering
2015-03-19 18:05:30,756 | 192.168.16.73 | INFO | User Logout | CIO Engineering
2015-03-19 18:05:34,723 | 192.168.16.73 | WARN | Login failed | cioS Engineering
2015-03-19 18:05:41,806 | 192.168.16.73 | INFO | Login successful | http://tomcat/testContext/rest/login
admin [ADMIN] 0 Health
2015-03-19 18:05:41,806 | 192.168.16.73 | INFO | Login successful | http://tomcat/testContext/rest/user
admin [ADMIN] 0 Ohters
2015-03-19 18:05:43,313 | 192.168.16.73 | INFO | User Logout | admin Others
2015-03-19 18:05:41,806 | 192.168.16.73 | INFO | Login successful | http://tomcat/testContext/rest/login
admin [ADMIN] 0 Health
2015-03-19 18:05:41,806 | 192.168.16.73 | INFO | Login successful | http://tomcat/rest/AED10/applications/2
/snapshots admin [ADMIN] 0 Engineering | Application - DATASTAGE
```



Note that the **rest URL**, **user role** and **delay time** options are only displayed for successful login messages.

Enabling DEBUG log mode for Active Directory/SAML authentication issues

By default, the log mechanism is not configured to provide any logging information to debug Active Directory/SAML authentication issues (see the relevant section in [User authentication](#)). If you have encountered issues activating Active Directory authentication, you can enable DEBUG log mode specifically to help resolve the problems you are having. To do so, you need to modify the following file with a text editor:

```
WAR 1.x
CATALINA_HOME\webapps\<dashboard>\WEB-INF\log4j2.xml

WAR 2.x
CATALINA_HOME\webapps\<dashboard>\WEB-INF\classes\log4j2-spring.xml

ZIP 2.x
<unpacked_zip>\configurations\log4j2-spring.xml
```

Uncomment the following sections as appropriate (this can be found at the end of the file):

```
<!-- Uncomment on LDAP access issues -->
<!--
<Logger name="org.springframework.security" level="debug">
    <AppenderRef ref="textout"/>
</Logger>
-->
<!-- Uncomment on SAML access issues -->
<!--
<Logger name="org.springframework.security.saml" level="debug" additivity="false">
    <AppenderRef ref="textout"/>
    </Logger>
    <Logger name="org.opensaml" level="debug" additivity="false">
    <AppenderRef ref="textout"/>
</Logger>
-->
```

Following any changes you make, **save the .xml file** and then **restart** your application server so that the changes are taken into account. Messages relating to Active Directory authentication will now be displayed in the log files.

Enabling DEBUG and SQL trace logs

By default, the log file will not contain any DEBUG or SQL trace logs. If you need to see these items in the log file, you can enable them as follows. Modify the following file with a text editor:

```
WAR 1.x
CATALINA_HOME\webapps\<dashboard>\WEB-INF\log4j2.xml

WAR 2.x
CATALINA_HOME\webapps\<dashboard>\WEB-INF\classes\log4j2-spring.xml

ZIP 2.x
<unpacked_zip>\configurations\log4j2-spring.xml
```

1.x

Locate the following section:

```
<Logger name="org.springframework.jdbc.core.JdbcTemplate" level="error" additivity="false">
    <AppenderRef ref="textout"/>
</Logger>
<Logger name="org.springframework.jdbc.core.StatementCreatorUtils" level="error" additivity="false">
    <AppenderRef ref="textout"/>
</Logger>
```

Change `level="error"` to `level="debug"` to enable DEBUG logs:

```
<Logger name="org.springframework.jdbc.core.JdbcTemplate" level="error" additivity="false">
    <AppenderRef ref="textout" />
</Logger>
<Logger name="org.springframework.jdbc.core.StatementCreatorUtils" level="error" additivity="false">
    <AppenderRef ref="textout" />
</Logger>
```

Additionally, if you want to see SQL trace logs as well as DEBUG information, change `level="debug"` to `level="TRACE"`:

```
<Logger name="org.springframework.jdbc.core.JdbcTemplate" level="TRACE" additivity="false">
    <AppenderRef ref="textout" />
</Logger>
<Logger name="org.springframework.jdbc.core.StatementCreatorUtils" level="TRACE" additivity="false">
    <AppenderRef ref="textout" />
</Logger>
```

2.x

Locate the following commented section:

```
<!-- <Logger name="org.springframework.jdbc.core" level="debug" additivity="false">
    <AppenderRef ref="textout" />
</Logger> -->
```

Uncomment this section to enable DEBUG logs:

```
<Logger name="org.springframework.jdbc.core" level="debug" additivity="false">
    <AppenderRef ref="textout" />
</Logger>
```

Additionally, if you want to see SQL trace logs as well as DEBUG information, change `level="debug"` to `level="TRACE"`:

```
<Logger name="org.springframework.jdbc.core" level="TRACE" additivity="false">
    <AppenderRef ref="textout" />
</Logger>
```

Following any changes you make, **save the .xml file** and then **restart** your application server so that the changes are taken into account.