

Logging mit Log4J

AID 074 DE



© 2014 ADITO Software GmbH

Diese Unterlagen wurden mit größtmöglicher Sorgfalt hergestellt. Dennoch kann für Fehler in den Beschreibungen und Erklärungen keine Haftung übernommen werden. Wir sind für Feedback zu den Themen, Inhalten, aber auch noch vorhandenen Fehlern dankbar und würden uns freuen, Ihre Meinung zu hören. Die in diesen Unterlagen enthaltenen Daten und Angaben, einschließlich URLs und anderer Verweise können ohne vorherige Ankündigung geändert werden. Alle in diesen Unterlagen aufgeführten Produkt- und Firmennamen sind unter Umständen Marken oder geschützte Zeichen der einzelnen Firmen. Ohne ausdrückliche schriftliche Einverständniserklärung der ADITO Software GmbH darf kein Teil dieses Dokumentes vervielfältigt oder in einer Datenverarbeitungsanlage gespeichert oder in diese eingelesen werden. Diese Einschränkung gilt unabhängig von Art und Weise der Datenerfassung.

Autor: FA, JK, KN. Version 10.1. Zuletzt geändert 19.09.2017

Version	Änderungen
10.1	Anpassung der Formatierungen
10.0	„log.log“ geändert zu „logging.log“
3.3	Letzter Stand vor Übernahme in Versionierung

Inhaltsverzeichnis

1.	Logging mit ADITO	4
1.1.	Log4J	4
2.	Konfiguration von log4J	5
2.1.	Aktivieren im Designer	5
2.2.	Konfigurationsdatei bearbeiten	6
3.	In der Praxis	9
3.1.	Ansprechen im ADITO-Code	9

1. Logging mit ADITO

Wie im Betriebshandbuch von ADITO online im Kapitel "Einstellungen im Preferences-Editor: Reiter Logging" eingesehen werden kann, bietet ADITO online vielfältige Möglichkeit der Logausgaben.

Üblicherweise wird die Logausgabe von ADITO online in eine Logdatei geschrieben (file logger), welche dann mit dem ADITO online Manager eingesehen werden kann. Alternativ können Logausgaben zur Laufzeit auch mit dem Konsolen-Logger oder dem Telnet-Logger ausgegeben werden.

1.1. Log4J

Bei log4J handelt es sich um ein Framework zum Loggen von Anwendungsdaten innerhalb von Java. Mit dieser Schnittstelle können Logging-Ausgaben von ADITO an alle Anwendungen übermittelt werden, welche Log4J unterstützen.

Die Logstufen, welche in der Log4J-Schnittstelle definiert werden können, können zur Laufzeit geändert werden (anders also als die normalen ADITO online Logging-Einstellungen). Grundsätzlich unterscheidet man zwischen:

- Dem **root logger**, dieser definiert Logging-Level und über welche Appender die Logging-Meldungen ausgegeben werden.
- Der **Appender** definiert den Ausgabekanal, z.B. Konsole, Datei, SMTP (für E-Mails) oder das System-Log.
- Das **Layout** definiert die Formatierung.



Details zu log4J finden Sie auf der offiziellen Website:
<http://logging.apache.org/log4j/2.x/>

2. Konfiguration von log4J

Damit die Konfiguration vorgenommen werden kann, muss die Konfigurationsdatei bei ADITO im gleichen Verzeichnis liegen wie die `log4j.jar`, also im Konfigurationsverzeichnis. Standardmäßig wird die Datei als Template im Konfigurations-Ordner `/config` von ADITO online ausgeliefert. Sie muss nur noch umbenannt werden, um zu funktionieren.

Wird log4J aktiviert und kein Dateiname für die Konfigurationsdatei angegeben, so erwartet das Logging diese Datei im Namen `log4j.configuration`.

2.1. Aktivieren im Designer

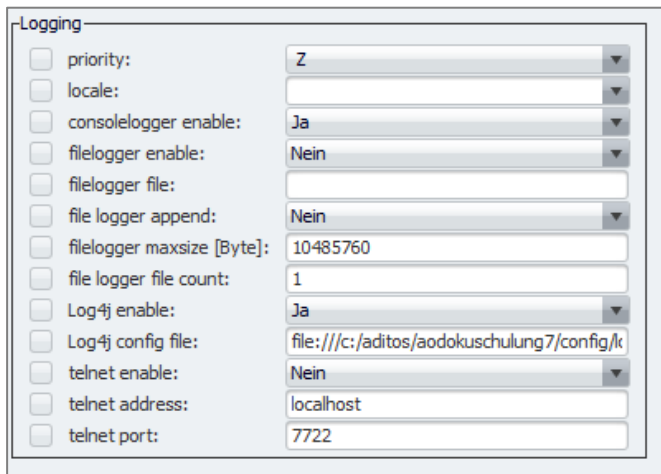
Um log4J zu aktivieren, muss der entsprechende Logger im Preferences-Editor des Designers aktiviert werden und die Datei im Verzeichnis vorliegen. Der Pfad zur Konfigurationsdatei wird immer als URL angegeben, z.B.

```
file:///C:/aditos/aodokusschulung6/config/log4j.configuration
```

Die Datei muss dabei denselben Namen tragen wie die entsprechende `log4j.jar`-Datei. Heißt die Datei beispielsweise `log4j-1.2.17.jar`, muss die Datei

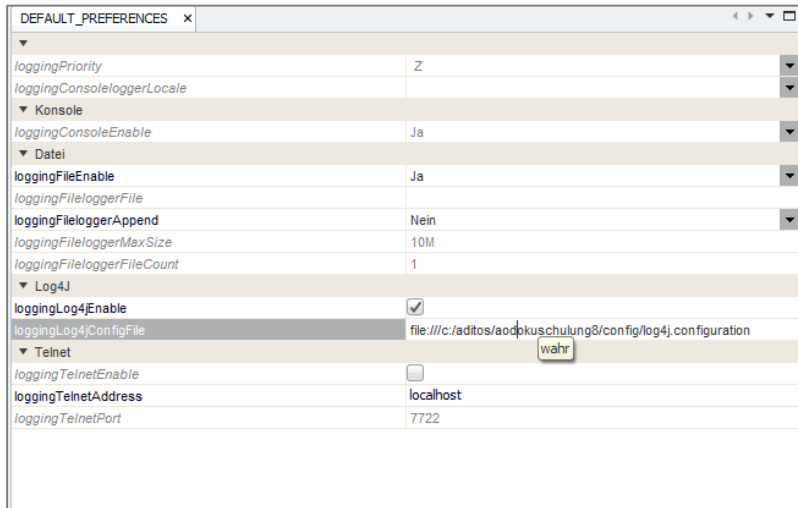
```
file:///C:/aditos/aodokusschulung6/config/log4j-1.2.17.configuration
```

benannt werden.



Option	Value
<input type="checkbox"/> priority:	Z
<input type="checkbox"/> locale:	
<input type="checkbox"/> consolelogger enable:	Ja
<input type="checkbox"/> filelogger enable:	Nein
<input type="checkbox"/> filelogger file:	
<input type="checkbox"/> file logger append:	Nein
<input type="checkbox"/> filelogger maxsize [Byte]:	10485760
<input type="checkbox"/> file logger file count:	1
<input checked="" type="checkbox"/> Log4j enable:	Ja
<input type="checkbox"/> Log4j config file:	file:///c:/aditos/aodokusschulung7/config/k
<input type="checkbox"/> telnet enable:	Nein
<input type="checkbox"/> telnet address:	localhost
<input type="checkbox"/> telnet port:	7722

(ADITO online 3.1)



(ADITO4)

2.2. Konfigurationsdatei bearbeiten

Die Template-Datei kann angepasst werden, um die gewünschten Logger zu aktivieren. Zuerst ist der Loglevel anzugeben. Standardmäßig ist der Level auf `DEBUG` eingestellt, und der Appender auf `Console`.

```
# Set root logger level to DEBUG and its only appender to Console.
log4j.rootLogger=DEBUG, Console
```

Folgende Level sind möglich und hierarchisch abgebildet. Die folgende Ordnung entspricht der Anzahl der Logmeldungen

ALL > TRACE > DEBUG > INFO > WARN > ERROR > FATAL > OFF



Gegebenenfalls muss pro Logger auch noch die Priorität für diesen Logger angegeben werden, z.B im File-Logger:

```
log4j.appender.FILE.Threshold=ERROR
```

Diese Priorität gilt dann nur für den jeweiligen Logger.

Man kann diesen so umstellen, dass das Logging auch auf das NT Evenhandling umgestellt wird:

```
# Set root logger level to DEBUG and its only appender to Console.
log4j.rootLogger=DEBUG, NT
```

Nun muss der entsprechende Knoten für den NT Logger noch angepasst werden:

Beispiel 1: NT Logger

```
# "NT" is set to be a NTEventLogAppender
log4j.appender.NT=org.apache.log4j.nt.NTEventLogAppender
```

```
log4j.appender.NT.source=Adito Online
log4j.appender.NT.server=127.0.0.1
log4j.appender.NT.layout=org.apache.log4j.PatternLayout
log4j.appender.NT.layout.ConversionPattern=%m
```



Für den NT Logger müssen die Dateien `NTEventLogAppender.dll` und `NTEventLogAppender.amd64.dll` vorliegen!

Soll die Logausgabe in eine Datei umgeleitet werden, kann der File Logger für Log4J verwendet werden.

Beispiel 2: File Logger

```
# Set root logger level to DEBUG and its only app
log4j.rootLogger=DEBUG, FILE

# "FILE" is set to be a FILEEventLogAppender
log4j.appender.FILE=org.apache.log4j.FileAppender
log4j.appender.FILE.File=C:/Users/f.pollner/Desktop/ADITO4/log.out
log4j.appender.FILE.source=Adito Onlin
log4j.appender.FILE.server=127.0.0.1
log4j.appender.FILE.layout=org.apache.log4j.PatternLayout
log4j.appender.FILE.layout.ConversionPattern=%m
log4j.appender.FILE.Threshold=ERROR
```

Beispiel 3: File Logger mit rollierendem Logfile

```
# Set root logger level to DEBUG and its only app
log4j.rootLogger=DEBUG, FILE

# "FILE" is set to be a FILEEventLogAppender
log4j.appender.FILE=org.apache.log4j.RollingFileAppender
log4j.appender.FILE.File=C:/Users/f.pollner/Desktop/ADITO4/log.out
log4j.appender.FILE.MaxFileSize=10000KB
log4j.appender.FILE.MaxBackupIndex=10
log4j.appender.FILE.source=Adito Onlin
log4j.appender.FILE.server=127.0.0.1
log4j.appender.FILE.layout=org.apache.log4j.PatternLayout
log4j.appender.FILE.layout.ConversionPattern=%m
```

```
log4j.appender.FILE.Threshold=ERROR
```

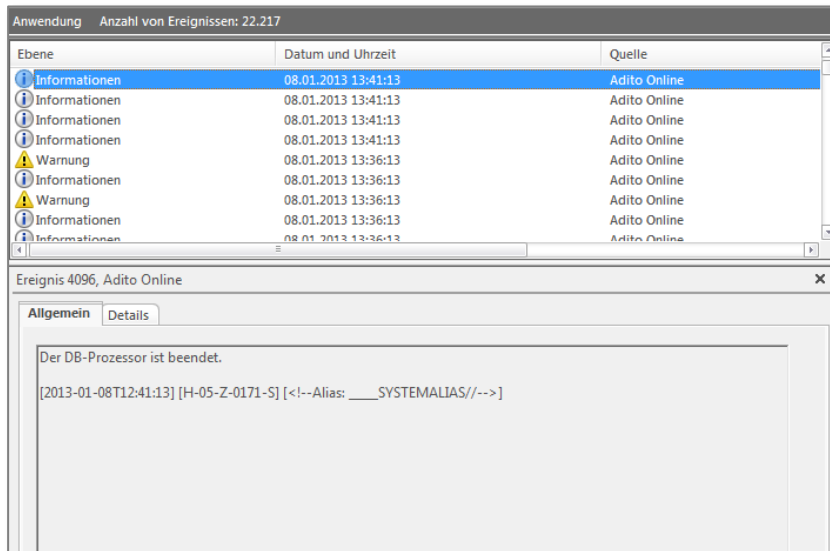
Nach Neustart des ADITO online Servers werden diese Änderungen aktiv.



Haben Sie einmalig den log4J-Logger aktiviert und ändern dann im laufenden Betrieb nur noch die Konfigurationsdatei, so ist kein Serverneustart mehr nötig.

3. In der Praxis

Wurde Log4J korrekt eingerichtet, so können die Meldungen ausgelesen werden. Wurde beispielsweise der NT Appender ausgewählt, so können in der Windows-Ereignisanzeige die Logmeldungen von ADITO online angezeigt werden.



3.1. Ansprechen im ADITO-Code

Um Logausgaben in ADITO anzugeben, werden die `logging.log`-Methoden verwendet. Diese können über den Parameter `ErrorID` angegeben werden.

Beispiel: Es werden folgende Logausgaben definiert:

```
logging.log("Nicht kritisch", logging.INFO);
logging.log("Kritisch", logging.ERROR);
```

In der normalen Logausgabe von ADITO erscheint dieser Code wie folgt:

```
R-37-Z-0001-S Projekt-Ausgabe. [ID 1] [->] Nicht kritisch
R-37-M-0003-S Projekt-Ausgabe. [ID 1] [->] Kritisch
```

Ist in Log4J die Priorität allerdings auf `ERROR` gestellt, wird nur folgendes geloggt:

```
Projekt-Ausgabe.

[2014-02-07T09:54:03] [R-37-M-0003-S] [ID 1] [<!--Kritisch/-->]
```



Der `logging.log`-Errorlevel ist ein anderer als der von Log4J. Zu den `logging.log`-Errorleveln in der `ErrorID` konsultieren Sie bitte das Betriebshandbuch.