

Anleitung lfa4uls - LogFileAdapter for ULS

lfa4uls ist ein einfacher Logfileadapter, mit dem es möglich ist, Logfiles zu überwachen, zu filtern und Aktionen auszuführen.

Der Logfileadapter überwacht eine Datei und führt für jede neue Zeile folgende Aktionen durch:

1. Wird die Zeile durch vreg oder vreg_file herausgefiltert? Wenn ja, dann auf nächste Zeile warten
2. Entspricht die Zeile einem Suchmuster reg0-reg99 oder aus reg_file0-reg_file99? Wenn ja, dann die entsprechende action<n> ausführen und auf neue Zeile warten.
3. Entspricht die Zeile dem Fehlermuster? Wenn ja, dann Fehleraktion ausführen und auf nächste Zeile warten.
4. Entspricht die Zeile dem Warnmuster? Wenn ja, dann Warnaktion ausführen und auf nächste Zeile warten.
5. Entspricht die Zeile dem Logmuster? Wenn ja, dann Logaktion ausführen und auf nächste Zeile warten.

Die Muster werden über Programmparameter oder in der Konfigurationsdatei festgelegt.

Programmaufruf:

```
lfa4uls [-D] [-a] [-c <conffile>] [-s|-S]
        [-v <reg>|-vf <vfile>] [-f <inputformat>] [-t <time>]
        [-e <ereg>|-ef <efile>|-ea] [-E <erroraction>]
        [-w <wreg>|-wf <wfile>|-wa] [-W <warnaction>]
        [-l <lreg>|-lf <lfile>|-la] [-L <logaction>] <path>

-a          Alle Zeilen betrachten
-c pfad     Konfigurationsdatei angeben
-s          Aktionen werden nur angezeigt, nicht ausgeführt
-S          wie -s, es wird nicht auf neue Zeilen gewartet
-v reg      Regexps, die nicht betrachtet werden sollen
-vf pfad    Regexps für -v aus Datei lesen
-f format   Format der Zeilen des Logfiles
-t time     Zeit in Sekunden, die auf neue Zeilen gewartet wird, 0 = 1/10 s
-e reg      Regexps, die die Fehleraktion auslösen
-ef pfad    Regexps für -e aus Datei lesen
-ea         alle Zeilen als Fehler betrachten
-E action   Aktion, die bei Fehlern ausgeführt wird
-w reg      Regexps, die die Warnaktion auslösen
-wf pfad    Regexps für -w aus Datei lesen
-wa         alle Zeilen als Warnung betrachten
-W action   Aktion, die bei Warnungen ausgeführt wird
-l reg      Regexps, die die Logaktion auslösen
-lf pfad    Regexps für -l aus Datei lesen
-la         alle Zeilen als Log betrachten
-L action   Aktion, die bei Log ausgeführt wird
<path>     der zu überwachenden Datei
```

Aussehen der Konfigurationsdatei:

```
# Konfigurationsdatei für lfa4uls
# Zu überwachende Datei
inputfile   = /var/log/messages

# Format der Logdatei
# folgende Codes können benutzt werden:
# %%    -> %
# %s*   -> beliebige Zeichenkette
# %s    -> ein Wort
# %t    -> Datum im Format: MMM TT hh:mm:ss (Monat Tag Stunde:Minute:Sekunde)
# %c    -> ein Zeichen
# %h    -> Host
```

```

# %q -> Quelle
# %W -> Meldung
# %D -> Datum im Format: TT.MM.JJ oder JJJJ-MM-TT
# %T -> Zeit im Format: hh:mm:ss
# %d -> Tag
# %m -> Monat
# %y -> Jahr
# %H -> Stunde
# %M -> Minute
# %S -> Sekunde
# %0 -> Variable 0
# %1 -> Variable 1
# ...
# %9 -> Variable 9
inputformat = %t %h %q: %W

# Datei mit den Ausschlusszeilen angeben
vreg_file = lfa4uls_v.lst

# Regexp für Ausschlusszeilen angeben
#vreg =

# Fehlermuster
# ereg_all auf 1, wenn alle Zeilen die Fehleraktion auslösen sollen
#ereg_all = 1

# Fehlermuster aus Datei
#ereg_file =

# Fehlermuster direkt angeben
ereg = ERROR

#Fehleraktion
# folgende Codes können verwendet werden:
# alle Codes des Inputfomates ohne %t, %s, %c
# %Y -> Jahr im Format JJJJ
# %b -> Monatsname
# %z -> die ganze Zeile
eaction = echo 'Error: %z'

# Warnmuster
# wreg_all auf 1, wenn alle Zeilen die Warnaktion auslösen sollen
#wreg_all = 1

# Warnmuster aus Datei
wreg_file = lfa4uls_w.lst

# Warnmuster direkt angeben
#wreg =
waction = echo 'Warnung: %z'

# Logmuster
# lreg_all auf 1, wenn alle Zeilen die Logaktion auslösen sollen
lreg_all = 1

# Logmuster aus Datei
#lreg_file =

# Logmuster direkt angeben
#lreg =

# Logaktion
laction = echo 'Logging: %z'

#
# Weitere Suchmusterdateien reg_file0-reg_file99 oder Suchmuster reg0-reg99
# mit den Aktionen action0-99.
# Zusätzlich können individuelle Eingabemuster bestimmt werden iformat0-iformat99
# Wenn zu einem Suchmuster kein Inputformat angegeben ist, dann wird
# das standard inputformat verwendet.

```

```
# Die Muster werden bei 0 beginnend in aufsteigender Reihenfolge verglichen.  
# Anschließend werden erst die ereg, wreg und lreg verglichen.  
# Die Zahlen sollten von 0 beginnend in aufsteigender Reihenfolge vergeben werden  
# denn Lücken würden das Programm unnötig verlangsamen.  
#  
#reg_file0 =  
reg0        = 12345  
action0     = echo 'Muster0: 1: %1, 2: %2, 3: %3, 4: %4'  
ifformat0   = %1|%2|%3|%4  
#  
#reg_file1 =  
#reg1       =  
#action1    =  
#ifformat1  =  
#
```