

TETRAcontrol Leitstelle/FEZ

Konfiguration der Ereignisverarbeitung



Übersicht

Mit der Ereignisverarbeitung kann TETRAcontrol auf eintreffende Ereignisse reagieren und Aktionen auslösen.

Ereignisse können sein:

- Status
- SDS
- Alarme (Callout)
- MQTT Daten

Folgende Aktionen können ausgelöst werden:

- SDS Versand
- Status Versand
- Alarm/Callout Versand
- Aufruf einer URL (GET oder POST)
- Ausführen einer Datei
- MQTT Publish

Einem Ereignis können auch mehrere Aktionen zugeordnet werden.

Die komplette Definition von Ereignissen und Aktionen wird in „.csv“ Dateien im Unterordner „Control“ gespeichert. Diese Dateien enthalten durch Semikolon („;“) getrennte Daten, die mit einem Text-Editor oder auch mit Excel oder OpenOffice Calc bearbeitet werden können.

Ereignisse definieren

Zuerst müssen die Ereignisse definiert werden, auf die TETRAcontrol reagieren soll. Der Aufbau der Datei „Events.csv“ ist wie folgt:

EventName;sISSI;dISSI;PID/Typ;Status;Text;MQTTtopic;MQTTval;ActionGroup;Action1;Para1;Para2;Para3

EventName	Eindeutiger Name für das Ereignis	
sISSI	Absender ISSI	
dISSI	Ziel ISSI/GSSI	
PID/Typ	SDS PID (z.B. 138) oder Typ (13 = Status)	
Status	Statuswert in Hex bei Typ=13	
Text	Text der SDS bzw. Statusklartext	
MQTTtopic	MQTT Topic auf das reagiert werden soll	MQTT Client muss aktiv sein!
MQTTval	MQTT Wert auf den reagiert werden soll	MQTT Client muss aktiv sein

ActionGroup	Aktionsgruppe die ausgelöst werden soll	
Action1	Einzelne Aktion die ausgeführt werden soll	
Para1 / 2 / 3	Werte die an die Aktionen übergeben werden	

Die Übereinstimmungen der einzelnen Felder werden als „Regular Expressions“ geprüft. Dies ermöglicht eine große Flexibilität bei der Definition. Es ist empfehlenswert sich ausführlich mit Regular Expressions auseinanderzusetzen. Im Internet gibt es viele gute Anleitungen und auch Dienste mit denen man die erstellen Expressions überprüfen kann, wie sie auf verschiedene Werte reagieren. Z.B.

<https://regex101.com/>

Beispiele:

```
Bsp1;6789000;1234567;13[07];;*Nürnberg.*;XXX;XXX;;AktionNr1
Status2;1234567;;13;8004;;XXX;XXX;;Torlauf
ALLmqtt;X;X;X;X;X;.*;.*;MQTT2;
Alarm1;;2690610;195;&0[25];;XXX;XXX;GrpMA12;MQAlarm2
```

Bsp1	6789000	1234567	13[07]		.*Nürnberg.*	XXX	XXX		AktionNr1
Erklärung:	Absender =6789000	Ziel=1234567	SDS Typen 130 und 137	Kein Status	Text Enthält „Nürnberg“	Mein MQTT	Kein MQTT	Keine Gruppe	AktionNr1 ausführen
Status2	123456[0-9]		13	8004		XXX	XXX		Torlauf
Erklärung:	Absender =1234567	Ziel=egal	Status (13)	Status 2	Text egal	Mein MQTT	Kein MQTT	Keine Gruppe	Torlauf ausführen
Alarm1		267800	195	&0[25]		XXX	XXXG	GrpMA12	MQAlarm2
Erklärung:	Absender=egal	Ziel=267800	Callout (195)	Subgruppen &02 + &05	Text egal	Mein MQTT	Kein MQTT	Gruppe „GrpMA12“ ausführen	„MQAlarm2“ ausführen

Aktionen definieren

Die auszuführenden Aktionen werden in der Datei „Actions.csv“ definiert. Der Aufbau der Datei ist wie folgt:

```
Typ;Name;Parameter1;Parameter2;Parameter3;Parameter4
```

Name ist ein eindeutiger Name mit dem die Aktion in der Events.csv oder in einer Gruppe referenziert wird.

Typ ist ein Wert zwischen 1 und 4 der die Art der Aktion definiert:

1	SDS Versenden
2	URL per HTTP/HTTPS aufrufen
3	Datei ausführen
4	MQTT Publish ausführen
5	Status Versenden (noch nicht implementiert)

Die Parameter sind abhängig von Typ:

1	Ziel ISSI	Text	PID	
2	URL	Benutzername	Passwort	POST Data

3	Dateiname (Vollständiger Dateiname incl. Endung).	Parameter	Arbeitsverzeichnis	Fenstermodus 0=Normal 1=Versteckt 2=Minimiert 3=Vollbild
4	MQTT Topic	Wert	QoS (0-2)	

Beim Typ SDS wird die PID numerisch eingegeben (130 = standard SDS, 137 = flash SDS, 195 = Callout). Um einen Status zu versenden ist der Typ SDS zu verwenden und als PID der Wert 13 einzutragen. Beim zu versenden Text ist dann der Status-Wert hexadezimal anzugeben (z.B. 800A für Status 8).

Bei Typ URL sind die Parameter „Benutzername“ und „Passwort“ optional und nur dann nötig, wenn eine Benutzerauthentifizierung nötig ist. Wenn der Parameter „POST Data“ leer ist, wird eine GET Anfrage durchgeführt.

Deim Typ „Datei ausführen“ muss sich der Dateiname im Arbeitsverzeichnis befinden. Falls nicht, kann der Dateiname mit vollem Pfad angegeben werden.

Bei den meisten Parametern können Variablen definiert werden, die beim Aufruf durch die jeweiligen Werte ersetzt werden:

\$TEXT\$	Text einer eingehenden SDS/Alarm bzw Statusklartext
\$ANAME\$	Absender Name (gemäß ISSI-Liste)
\$ASID\$	Absender Subscriber Identity (ISSI)
\$ZNAME\$	Ziel Name (gemäß ISSI/GSSI Listen)
\$ZSID\$	Ziel Subscriber Identity (ISSI bzw GSSI)
\$SUB\$	Sub-Gruppe (nur bei Callout)
\$PRIO\$	Priorität / Severity (bei Callout)
\$STATUS\$	Statuswert in Hex (bei Status)
\$STATUSFMS\$	Statuswert als ‚FMS-Status‘ (1-9)
\$WERT1\$	Parameter 1 aus der Event-Konfiguration
\$WERT2\$	Parameter 2 aus der Event-Konfiguration
\$WERT3\$	Parameter 3 aus der Event-Konfiguration

Beispiele:

```
2;Divera1;https://www.divera247.com/api/alarm?accesskey=AEpxxxx&text=$TEXT$&ric=124711;;;
2;Divera2;https://www.divera247.com/api/alarm?accesskey=MeinKey2xxxx&text=$TEXT$&ric=124712;;;
2;DiveraSt;https://www.divera247.com/api/fms?accesskey=MeinKey&status_id=$STATUSFMS$&vehicle_issi=$ASID$;;;
4;AlarmSteuerung;IO/cmnd/AlarmSteuerung/Power1;1;1;
4;MQAlarmISM;Alarm/MK/Mitte/SM;Alarm für die Feuerwehr Mitte $TEXT$;;
1;SDSt1a;8761483;Tor ELW ist AUF!;137;
2;T.FF1SP;https://api.telegram.org/bot6912326:AAEkvqlx3f48ptLc/sendMessage?chat_id=-381232&text=$ANAME$+$TEXT$;;;
2;T.FF1CO;https://api.telegram.org/bot6912326:AAEkvqlx3f48ptL/sendMessage?chat_id=-283496&text=$WERT1$: $ANAME$+$ZNAME$+$SUB$+$TEXT$;;;
3;Dateil;test.bat;"$ANAME$" "$TEXT$";C:\temp\;1
```

Aktionen gruppieren

Wenn mehrere Aktionen mit einem Ereignis ausgelöst werden sollen, dann können diese in einer Gruppe zusammengefasst werden.

Hierzu ist die Datei „ActionGroups.csv“ mit folgendem Inhalt zu nutzen:

```
Name;Action1;Action2;Action3;....
```

Beispiel:

```
GrpMAL2;AlarmSteuerung;SDSt1a;T.FF1CO;Divera2
```