H T T P   A P I   S p e c i f i c a t i o n

Benning TLS series inverters

Version 1.2

Benning GmbH & Co KG
Münsterstraße 135 – 137
46397 Bocholt

1 Table of content

2 Introduction

Inverter devices of BENNING TLS series are equipped with an Ethernet interface that allows accessing the internal webserver. Most of the internal measurement data is accessible via that webserver using HTTP requests to specific CGI handlers. Data is delivered in JSON (https://en.wikipedia.org/wiki/JSON) format which is easy to read and parse automatically. There are JSON libraries ready to use for most common programming languages.

Most of the data is read-only without authentication which is sufficient for monitoring and energy management. For changing settings, a login has to be performed and a HTTP session cookie needs to be managed by the client.

3 Reading data

Measurement and status information may be read from the devices using a CGI request as followes

```
http://<IP-Address>/getentry.cgi?&oid=<Object ID>
```

The passed parameter “Object ID” (OID) specifies the requested information. For a list of all supported OID see at the end of the document.

Basically every measurement value, status information of configuration setting is addressable via such object identifier.

Upon success (Object exists and is readable), the webserver will return the object in JSON notation as follows:

```
{
"oid" : <Object ID>,
"label" : "<object label>",
"type" : "<data type>",
"val" : "<value>",
"uitext" : "<user interface text>",
"savetype" : <savetype >,
"factor" : <scaling factor>,
"unit" : "<physical measurement unit>",
"portal" : <Portal submission flag>,
"AutoZero" : <Autozero flag>,
"fb" : <Fieldbus flag>
}
```

The elements provide the following information (some element are internally used only, and should be ignored):

- **ObjectID**: unique identifier of the data object
- **label**: unique human readable identifier of the requested object
- **type**: Datatype of the result. The following types are supported:
  - 'F' -> Float
  - 'L' -> Int64
  - 'i' -> Int32
  - 'w' -> Int16
  - 'b' -> Int8
  - 'S' -> String (UTF-8 coded)
  - 'B' -> Boolean
  - 'I' -> Ip-Address
- **val**: actual value of the requested value according to the given datatype.
- **uitext**: The name/desciption of the object when presented in a user interface (e.g. web interface, UTF-8 coded)
- **savetype**: archiving type for that value, used internally for determining if and how to archive that value. May be ignored.
- **factor**: scaling factor for physical units. The value needs to be multiplied with this factor to get the result in given unit. Used e.g. to present Wh energy values as kWh.

- **unit**: The physical unit of the value as string (UTF-8 coded). The values need to be multiplied with the variable “factor” to present it using this unit.

- **AutoZero**: whether to set a value to zero when not updated any more. Used internally, may be ignored.

- **fb**: determines if this object is broadcasted via fieldbus. Only broadcasted values are available at the portal and when reading the status of remote inverter from another inverter.

In case of an error, e.g. when the requested object is not available, no JSON object is returned by an errorcode like “-1”.

3.1 Examples

3.1.1 Reading the serial number:

As documented in the OID list below, the Serial Number has the OID 15. To read the serial number, the following request is performed:

Request:

http://192.168.1.2/getentry.cgi?&oid=15

Result:


```
{
   "oid" : 15,
   "label" : "UserConfig.Inverter.Info.SerialNumber.alphanumeric",
   "type" : "S",
   "val" : "106000123456",
   "uitext" : "Serial number",
   "savetype" : 1,
   "factor" : 1.000000,
   "unit" : "",
   "portal" : 1,
   "AutoZero" : 0,
   "fb" : 0
}
```


(Characters made **bold** to improve readability)

3.1.2 Reading the AC-Power:

The AC Power has ObjectID 11365, so to read its value, the following request has to be performed

Request:

```
http://192.168.1.2/getentry.cgi?&oid=11365
```

Result:


```
{
"oid" : 11365,
"label" : "SystemState.Global.Measurement.AcPowerTotal",
"type" : "i",
"val" : "9145",
"uitext" : "AC Power",
"savetype" : 2,
"factor" : 1.000000,
"unit" : "W",
"portal" : 0,
"AutoZero" : 1,
"fb" : 1
}
```


So the concerned system currently delivers 9.145 kW power.

3.1.3 Reading the inverter temperature:

The object ID of the temperature is 11470.

Request:

```
http://192.168.1.2/getentry.cgi?&oid=11470
```

Result:


```
{
  "oid" : 11470,
  "label" : "SystemState.Global.Measurement.TemperatureInverter",
  "type" : "i",
  "val" : "489",
  "uitext" : "Inverter temperature",
  "savetype" : 2,
  "factor" : 0.100000,
  "unit" : "°C",
  "portal" : 2,
  "AutoZero" : 1,
  "fb" : 1
}
```


In this measurement, the “Scaling factor” is 0.1 and the datatype is 32 bit integer. For getting the temperature in the given unit “°C”, the value hast to be multiplied by the factor 0.1

So the resulting temperature is 489 * 0.1 [°C] = 48.9 [°C]

4 Querying multiple values at once

Instead of reading a single value it is also possible to query a set of values with a single request. This is useful when many items are polled as it is much faster then sequentially polling single values.

The command for reading multiple values at once is as follows: (Caution: notice the changed command getentr**ies**.cgi and die changed parameter oid**s**!)

Request:

```
http://<IP>/getentries.cgi?oids=<OID1>[,<OID2>][,...]
```

Example

```
http://192.168.1.2/getentries.cgi?oids=11365,11470
```

Result:

The requested OIDs are returned as an array of JSON objects:

```
[
{<ENTRY1>},
{<ENTRY2>},
{<ENTRY3>},
, …
]
```

Example for the above request:


```
[
{
"oid" : 11365,
"label" : "SystemState.Global.Measurement.AcPowerTotal",
"type" : "i",
"val" : "1746",
"uitext" : "AC Power",
"savetype" : 2,
"factor" : 1.000000,
"unit" : "W",
"portal" : 0,
"AutoZero" : 1,
"fb" : 1
},
{
"oid" : 11470,
"label" :
"SystemState.Global.Measurement.TemperatureInverter",
"type" : "i",
"val" : "343",
"uitext" : "Inverter temperature",
"savetype" : 2,
"factor" : 0.100000,
"unit" : "°C",
"portal" : 2,
"AutoZero" : 1,
"fb" : 1
}
]
```


The dataformat of the single entries is identical to the format when quering single items – see specification there.

In case of an error – e.g. when querying a non-existent OID, a “-1” instead of the result error is returned.

5 Reading single values of group-devices (remote fieldbus device access)

Most measurement values are available for reading from remote inverter devices connected to the same fieldbus.

This requires that the remote fieldbus address is passed as a parameter when performing the request.

```
    http://<IP-Address>/getentry.cgi?&oid=<ObjectID>&fbAddr=<Fieldbus-Address>
```

Reading for example the system name (OID 10) of the fieldbus device with address 8 is performed as follows:

Request:

```
http://192.168.1.2/getentry.cgi?&oid=10&fbAddr=8
```

Result:


```
{
"oid" : 10,
"label" : "UserConfig.Inverter.Info.SystemName",
"type" : "S",
"val" : "PVINV108",
"uitext" : "System name",
"savetype" : 1,
"factor" : 1.000000,
"unit" : "",
"portal" : 1,
"AutoZero" : 0,
"fb" : 1
}
```


So the device name of the requested inverter is “PVINV108”.

6 Writing data

Writing data is possible for some configuration settings. Before writing, a session has to be established by authenticating with username and password. The request for logging in as follows:

```
GET http://<IP>/login.cgi?name=<user>&pass=<Password>
```

Example (with default password):

```
http://192.168.1.2/login.cgi?name=admin&pass=benning
```

Upon successful authenticaton, the inverter answers with„OK“, and delivers an HTTP session cookie containing the session ID: `BENNINGSESSID=1308438654`.

This cookie needs to be submitted with all following requests.

Please keep in mind, that only some configuration parameters are configurable, due to technical or legal/safety requirements. Writing configuration settings may lead to reduced yield, making the inverter unreachable via network or other malfunction. Please make sure that only correct settings are being written to the settings.

Writing data is possible after successful authentication by calling a CGI handler via HTTP. This handler need 3 parameters:

- the ObjectID to write
- the datatype to write
- the value to write

The syntax is:

GET http://**<IP-Adress>**/setentry.cgi?&oid=**<ObjectID>**&type**=<DATATYPE>**&val=**<VALUE>**

Please note, that writing data for remote fieldbus devices is currently not supported. Only data of the directly accessible device running the web-interface may be written.

Example: setting the fieldbus address (INT8 value with ObjectID 690) to “10”:

http://192.168.1.2/setentry.cgi?&oid=690&type=b&val=10.

Upon success, the new value is returned in JSON format:

```
{
        "oid" : 690,
        "label" :
        "UserConfig.Interfaces.Fieldbus.Address",
        "type" : "b",
        "val" : "10",
        "uitext" : "Fieldbus address",
        "savetype" : 1,
        "factor" : 1.000000,
        "unit" : "",
        "portal" : 1,
        "AutoZero" : 0,
        "fb" : 1
}
```

If an error occurred writing the Object, an errorcode like “-1” is returned

7 List of readable data objects

The following Objects IDs are available for reading via the webinterface:

OID	Label	Comment	Type	Unit	Name
11305	SystemState.Global.Measurement.ACFrequency	measured grid frequency in Hz; sent from Inverter-Controller	F	Hz	grid AC frequency
11306	SystemState.Global.Measurement.ACFrequencyMin	measured grid min and max frequency in Hz; sent from Inverter-Controller	F	Hz	grid AC frequency min
11307	SystemState.Global.Measurement.ACFrequencyMax		F	Hz	grid AC frequency max
11310	SystemState.Global.Measurement.AcCurrentL1	AC current phase 1 in [mA]; sent from Inverter-Controller	i	A	AC current L1
11315	SystemState.Global.Measurement.AcCurrentL2	AC current phase 2 in [mA]; sent from Inverter-Controller	i	A	AC current L2
11320	SystemState.Global.Measurement.AcCurrentL3	AC current phase 3 in [mA]; sent from Inverter-Controller	i	A	AC current L3
11325	SystemState.Global.Measurement.AcCurrentMaxL1	maximum AC current phase 1 in [mA]; sent from Inverter-Controller	i	A	AC current L1 max
11327	SystemState.Global.Measurement.AcCurrentMaxL2	maximum AC current phase 2 in [mA]; sent from Inverter-Controller	i	A	AC current L2 max
11329	SystemState.Global.Measurement.AcCurrentMaxL3	maximum AC current phase 3 in [mA]; sent from Inverter-Controller	i	A	AC current L3 max
11330	SystemState.Global.Measurement.AcCurrentMinL1	minimum AC current phase 1 in [mA]; sent from Inverter-Controller	i	A	AC current L1 min
11333	SystemState.Global.Measurement.AcCurrentMinL2	minimum AC current phase 2 in [mA]; sent from Inverter-Controller	i	A	AC current L2 min

11339	SystemState.Global.Measurement.AcCurrentMinL3	minimum AC current phase 3 in [mA]; sent from Inverter-Controller	i	A	AC current L3 min
11340	SystemState.Global.Measurement.DcCurrentMppTracker1	DC current MPP tracker 1 in [mA]; sent from Booster-Controller	i	A	DC current MPP tracker 1
11350	SystemState.Global.Measurement.DcCurrentMppTracker2	DC current MPP tracker 2 in [mA]; sent from Booster-Controller	i	A	DC current MPP tracker 2
11360	SystemState.Global.Measurement.DcCurrentMppTracker3	DC current MPP tracker 3 in [mA]; sent from Booster-Controller	i	A	DC current MPP tracker 3
11365	SystemState.Global.Measurement.AcPowerTotal	AC Power total in [W]; sent from Inverter-Controller as sum of the single phases L1+L2+L3	i	W	AC Power
11367	SystemState.Global.Measurement.ExternalS0Power	external counted power, calculated from S0 input pulses energy (derivative), displayed in [W]; calculated by Interface-Controller from S0counter derivative	i	W	external power counter
11368	SystemState.Global.Measurement.SelfConsumptionPower	self consumption power, displayed in [W]; minumum of S0 (self consumption) power and inverter power	i	W	self consumption power
11369	SystemState.Global.Measurement.FeedInPower	real feed in power (AC output minus self consumption power), displayed in [W]	i	W	feed-in power

11370	SystemState.Global.Measurement.AcPowerL1	AC Power on Phase L1 in [W]; sent from Inverter-Controller	i	W	AC Power L1
11380	SystemState.Global.Measurement.AcPowerL2	AC Power on Phase L2 in [W]; sent from Inverter-Controller	i	W	AC Power L2
11390	SystemState.Global.Measurement.AcPowerL3	AC Power on Phase L3 in [W]; sent from Inverter-Controller	i	W	AC Power L3
11395	SystemState.Global.Measurement.DcPowerTotal	DC Power total in [W]; sent from Booster-Controller as sum of the single strings tracker1+tracker2+tracker3	i	W	DC Power
11397	SystemState.Global.Measurement.DcPowerAvailable	the maximum possible availabe DC generator power. This is determined by multiplying the irradiation sensor resulte [W/m²] with the generator area size [m²].	i	W	Available DC Power
11400	SystemState.Global.Measurement.DcPowerMppTracker1	DC Power on MPP tracker 1 in [W]; sent from Booster-Controller	i	W	DC Power MPP tracker 1
11410	SystemState.Global.Measurement.DcPowerMppTracker2	DC Power on MPP tracker 2 in [W]; sent from Booster-Controller	i	W	DC Power MPP tracker 2
11420	SystemState.Global.Measurement.DcPowerMppTracker3	DC Power on MPP tracker 3 in [W]; sent from Booster-Controller	i	W	DC Power MPP tracker 3
11430	SystemState.Global.Measurement.AcReactivePower	AC reactive power (Blindleistung) in [Var]; sent from Inverter-Controller	i	Var	AC reactive power
11450	SystemState.Global.Measurement.AcApparentPower	AC apparent power (Scheinleistung) in [VA]; sent from Inverter-Controller	i	VA	AC apparent power
11460	SystemState.Global.Measurement.TemperatureHeatSink	Heat sink temperature in 0.1 [°C]; sent from Interface-Controller	i	°C	Heat sink temperature
11470	SystemState.Global.Measurement.TemperatureInverter	inverter temperature (avrg. of all sensors) in 0.1 [°C]; sent from Interface-Controller	i	°C	Inverter temperature
11480	SystemState.Global.Measurement.AcVoltageL1	AC Voltage (mains) on Phase L1 in [V]; sent from Inverter-Controller	F	V	AC Voltage L1
11490	SystemState.Global.Measurement.AcVoltageL2	AC Voltage (mains) on Phase L2 in [V]; sent from Inverter-Controller	F	V	AC Voltage L2
11500	SystemState.Global.Measurement.AcVoltageL3	AC Voltage (mains) on Phase L3 in [V]; sent from Inverter-Controller	F	V	AC Voltage L3

11510	SystemState.Global.Measurement.AcVoltageL1Max	maximum AC Voltage on Phase L1 in [V]; sent from Inverter-Controller	F	V	AC voltage L1 max
11520	SystemState.Global.Measurement.AcVoltageL2Max	maximum AC Voltage on Phase L2 in [V]; sent from Inverter-Controller	F	V	AC voltage L2 max
11530	SystemState.Global.Measurement.AcVoltageL3Max	maximum AC Voltage on Phase L3 in [V]; sent from Inverter-Controller	F	V	AC voltage L3 max
11540	SystemState.Global.Measurement.AcVoltageL1Min	minimum AC Voltage on Phase L1 in [V]; sent from Inverter-Controller	F	V	AC voltage L1 min
11550	SystemState.Global.Measurement.AcVoltageL2Min	minimum AC Voltage on Phase L2 in [V]; sent from Inverter-Controller	F	V	AC voltage L2 min
11560	SystemState.Global.Measurement.AcVoltageL3Min	minimum AC Voltage on Phase L3 in [V]; sent from Inverter-Controller	F	V	AC voltage L3 min
11570	SystemState.Global.Measurement.DcVoltageMppTracker1	DC Voltage on MPP tracker 1 in [V]; sent from Booster-Controller	F	V	MPP tracker 1 DC voltage
11580	SystemState.Global.Measurement.DcVoltageMppTracker2	DC Voltage on MPP tracker 2 in [V]; sent from Booster-Controller	F	V	MPP tracker 2 DC voltage
11590	SystemState.Global.Measurement.DcVoltageMppTracker3	DC Voltage on MPP tracker 3 in [V]; sent from Booster-Controller	F	V	MPP tracker 3 DC voltage
11600	SystemState.Global.Measurement.DcVoltageMppTracker1 Max	Maximum DC Voltage on MPP tracker 1 in [V]; sent from Booster-Controller	F	V	MPP tracker 1 DC voltage max
11610	SystemState.Global.Measurement.DcVoltageMppTracker2 Max	Maximum DC Voltage on MPP tracker 2 in [V]; sent from Booster-Controller	F	V	MPP tracker 2 DC voltage max
11620	SystemState.Global.Measurement.DcVoltageMppTracker3 Max	Maximum DC Voltage on MPP tracker 3 in [V]; sent from Booster-Controller	F	V	MPP tracker 3 DC voltage max
11650	SystemState.Global.OperatingStatus	operating status (standby, feedin, failure..); sent from Interface-Controller	i		operating status
11660	SystemState.Global.Measurements.LeakageCurrent	leakage current measured by the AFI [mA]; sent from Supervision-Controller	i	mA	leakage current
11670	SystemState.Global.Measurements.RadiationSensor	irradiation measured by external sensor [W/m^2]; sent from Interface-Controller	F	W/m ^2	external irradiation sensor
11680	SystemState.Global.Measurements.TemperatureSensor	temperature measured by external by external sensor [0.1 °C]; sent from Interface-Controller	i	°C	external temperature sensor

11750	SystemState.Global.Measurements.EarthInsulationResistance	Earth leakage insulation resistance in Ohm; sent from Supervision-Controller	i	Ohm	Earth Insulation Resistance
11760	SystemState.Global.Measurements.DeltaEtotal	Delta E_total: Energy produced since the last transmission of data to the protocol	L	kWh	total Energy produced since last portal transmission
11762	SystemState.Global.Measurements.Etoday	Delta Energy today : Energy produced today	L	kWh	Energy today
11764	SystemState.Global.Measurements.Emonth	Delta Energy today : Energy produced this month	L	kWh	Energy produced this month
11766	SystemState.Global.Measurements.Eyear	Delta Energy today : Energy produced this year	L	kWh	Energy produced this year
11767	SystemState.Global.Measurements.ESelfToday	self consumption Energy today : self consumed Energy today	L	kWh	Self consumption energy today
11768	SystemState.Global.Measurements.ESelfMonth	self consumption Energy this month : self consumed Energy this month	L	kWh	Self consumption energy this month
11769	SystemState.Global.Measurements.DeltaESelfConsumption	Delta of the self consumption energy since last portal transmission	L	kWh	self consumed energy since last portal transmission
19000	SystemState_persistent.Global.Measurement.E_total	Corrected/scaled total amount of energy produced by this inverter [Wh], displayed in [kWh]; calculated by Interface-Controller from E_total_uncorrected*Correction_factor + Correction_offset	L	kWh	total Energy produced
19010	SystemState_persistent.Global.OperatingHoursTotal	total operating hours of the inverter (frontpanel active, interface controller running) [h]; sent in secs (1/3600 h); sent from Interface-Controller	i	h	Total operating hours
19020	SystemState_persistent.Global.OperatingHoursFeedinIn	total feed-in hours of the inverter [h]; sent in secs (1/3600 h); sent from Inverter-Controller	i	h	feed-in operating hours

19035	SystemState_persistent.Global.Measurement.ExternalS0Energy	external counted energy, calculated from S0 input pulses, displayed in [kWh]; calculated by Interface-Controller from S0counter 1 * S0 rate [pulses/kWh]	L	kWh	external energy counter
19037	SystemState_persistent.Global.Measurement.SelfConsumptionEnergy	external self consumption energy [kWh]; calculated by Interface-Controller from self consumption power	L	kWh	self consumption energy counter

Additional Data, **only for energy storage systems (ESS)**:

12000	SystemState.Global.Measurement.BatteryVoltage	Measured battery voltage in [V]	F	V	Battery voltage
12010	SystemState.Global.Measurement.BatteryCurrent	Measured battery current in [A] when current is negative, the battery is discharged when current is positive, the battery is getting charged	F	A	Battery current
12015	SystemState.Global.Measurement.BatteryChargeCurrent	The measured battery discharge current in A. Calculated by interface controller based on BatteryCurrent	F	A	Battery charge current
12020	SystemState.Global.Measurement.BatteryDischargeCurrent	The measured battery charge current in A. Calculated by interface controller based on BatteryCurrent	F	A	Battery discharge current
12025	SystemState.Global.Measurement.BatteryPower	Measured battery power when positive, energy is stored into battery when negative, energy is taken from the battery	i	W	Battery power
12030	SystemState.Global.Measurement.BatteryChargingPower	Battery charging power,	i	W	Battery charging power
12035	SystemState.Global.Measurement.BatteryDischargingPower	Battery discharging power	i	W	Battery discharging power
12040	SystemState.Global.Measurements.BatteryTemperature	battery temperature [0.1 °C]	i	°C	Charger battery temperature
12098	SystemState.DcDc.ChargedEnergyToday	Amount of energy charged today by this DCDC converter	F	kWh	DCDC charged energy today
12099	SystemState.DcDc.DischargedEnergyToday	Amount of energy discharged today by this DCDC converter	F	kWh	DCDC discharged energy today
12210	SystemState.GlobalMeasurement.BMS.SoC	The State-of-charge of the battery, submitted by the BMS	F	%	BMS battery SoC
12220	SystemState.GlobalMeasurement.BMS.ChargedEnergyToday	Energy stored in battery today	F	kWh	Battery charged energy today
12225	SystemState.GlobalMeasurement.BMS.DischargedEnergyToday	Energy discharged/taken from battery today	F	kWh	Battery discharged energy today
12100	SystemState.MeasurementBox.VoltageL1	external grid measurement box: voltage L1	F	V	Grid Voltage L1
12105	SystemState.MeasurementBox.VoltageL2	voltage L2	F	V	Grid Voltage L2
12110	SystemState.MeasurementBox.VoltageL3	voltage L3	F	V	Grid Voltage L3
12120	SystemState.MeasurementBox.CurrentL1	current L1	F	A	Grid Current L1

12125	SystemState.MeasurementBox.CurrentL2	current L2	F	A	Grid Current L2
12130	SystemState.MeasurementBox.CurrentL3	current L3	F	A	Grid Current L3
12140	SystemState.MeasurementBox.PowerL1	power L1	F	W	Grid Power L1
12145	SystemState.MeasurementBox.PowerL2	power L2	F	W	Grid Power L2
12150	SystemState.MeasurementBox.PowerL3	power L3	F	W	Grid Power L3
12160	SystemState.MeasurementBox.GridPower	overall power taken (negative) or fed into (positive) the grid	F	W	Grid power
12165	SystemState.MeasurementBox.GridFeedinPower	Absolute value of power fed into the grid (always positive) When power is taken from the grid, this value is zero	F	W	Grid feedin power
12171	SystemState.MeasurementBox.GridObtainedEnergyTotal	Absolute energy (sum) obtained from the grid overall This value is counted up whenever energy is obtained from the grid. It is never reset and never decreased.	F	kWh	Obtained grid energy total
12173	SystemState.MeasurementBox.GridFeedinEnergyTotal	Absolute energy (sum) fed into the grid overall This value is counted up whenever energy is fed into the grid. It is never reset and never decreased.	F	kWh	Feedin grid energy total

8 Reading the events logbook

The inverter keeps an internal event logbook which can be read via the web interface. The CGI command is specified as follows:

http://<IP>/geteventlog.cgi?&start=<start-unixtimestamp>&end=<end-unixtimestamp>

By passing the parameters, a timespan is specified in which the events are queried:

- start-unixtimestamp: UNIX timestamp (seconds since 1.1.1970) since when the events shall be read from the eventlog
- end-unixtimestamp: UNIX timestamp (seconds since 1.1.1970) until when the events shall be read from the logbook

Example:

http://192.168.1.2/geteventlog.cgi?&start=0&end=1383751585

The webserver then answer with an JSON array of events:


```
[
      {
      "time" : 1383578619,
      "milliSecs" : 519,
      "id" : 60,
      "text" : "Firmware Update pending",
      "level" : 2,
      "action" : 1,
      "sync" : 2
      },
      {
       …
      },
      …
]
```


Description of the datafields:

- time: unix timestamp of the event (in seconds since 1.1.1970)
- milliSecs: internal millisecond counter when the event was processed (reserved for internal use)
- id: event identifier
- text: textual description of the event. Language according to the selected language of the webinterface.
- level: Severity of the event:

```
0 Trace   (lowest severity)     reserved
1 debug                         reserved
2 info                          informational event
3 warning                       warning event
4 error                         error event
5 fatal   (highest severity)    fatal error event
```

- action: info whether the event was coming/active(1) oder has gone/became inactive(0)
- sync: synchronisation mode, reserved for internal use