Motion control with EPICS

Torsten Bögershausen

Motion Control and Automation Group

www.europeanspallationsource.se

motorRecord

- EPICS has a complete package for motors:
  - record(motor)
  - Common code in generic driver (model 3)
  - Specific drivers for different controllers
- https://github.com/epics-modules/motor
- Many changes/improvements:
  https://bitbucket.org/
  europeanspallationsource/m-epics-motor

5 important things in record

- Which position should the motor be ?
- At which position is the motor ?
- Serious problems (what is this ?)
- What is the motor doing ?
- Stop the motor

4 most important fields

- EPICS record name: (P)(M)
- Fields belonging to the record:
  - Where should the motor be ? (P)(M).VAL
  - Where is the motor ? (P)(M).RBV
  - (Serious) problems (P)(M).STAT
  - Stop the motor (P)(M).STOP

Extra PV

- The (P)(M)-MsgTxt PV:
  Shows additional information:
  ’E: Controller error 4650’
  ’I: Both limit switches’
  ’I: Homing’

More important fields

- Read only fields:
  - (P)(M).MOVN Motor moving
  - (P)(M).DMOV Movement done
    (After backlash and delay)
  - (P)(M).MISS Position not reached
  - (P)(M).LVIO (Soft) Limit Violation
  - (P)(M).HLS, .LLS High/Low
    limit switch reached

Move to a position

- Read/Write fields:

  - (P)(M).VELO Velocity
  - (P)(M).ACCL Acceleration time (seconds)
  - (P)(M).VAL Target position

Move with constant velocity

- Read/Write fields:

  - (P)(M).JVEL Velocity
  - (P)(M).JAR Acceleration (mm/sec²)
  - (P)(M).JOGF Move forward
  - (P)(M).JOGR Move reverse

Homing

- Read/Write fields:
  - (P)(M).HOMF Home (forward)
  - (P)(M).HOMR Home (reverse)

- (Today, both do the same)
  All homing is configured in TwinCAT (nearly all).

Connecting EPICS to TwinCAT

Torsten Bögershausen

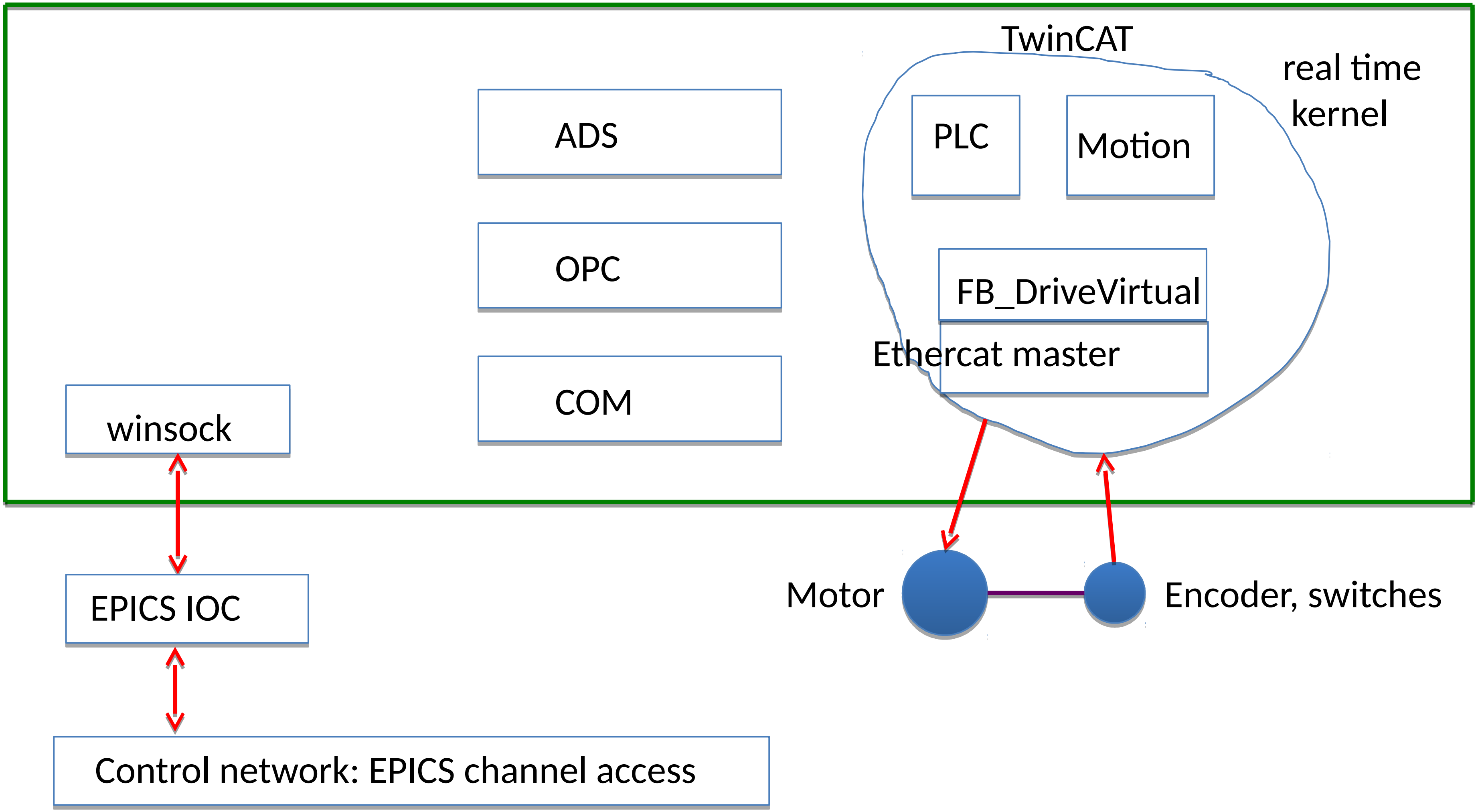
SW Engineer

Motion Control and Automation Group

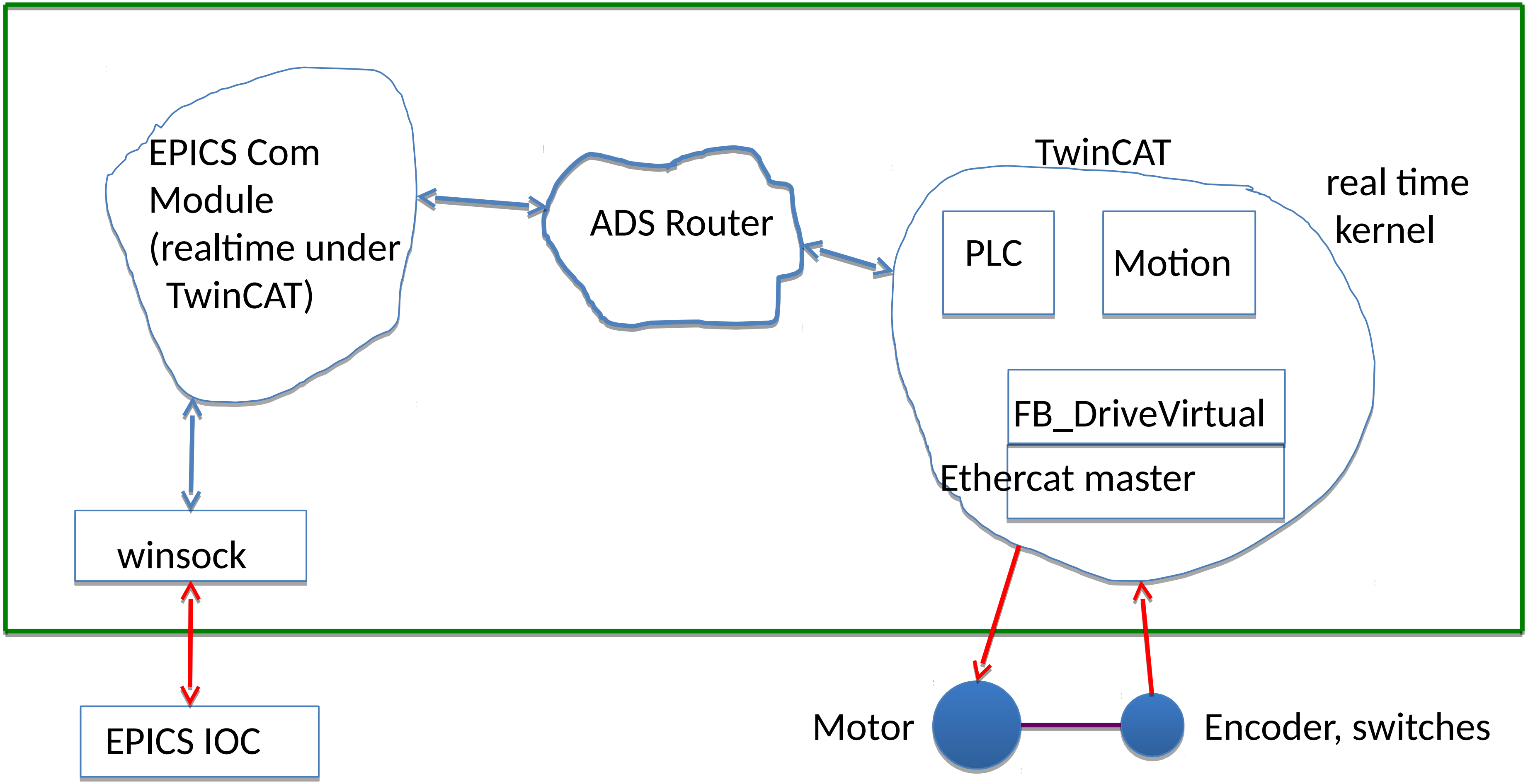
www.europeanspallationsource.se

2015-05-19

Connections to TwinCAT



EPICS Com Module

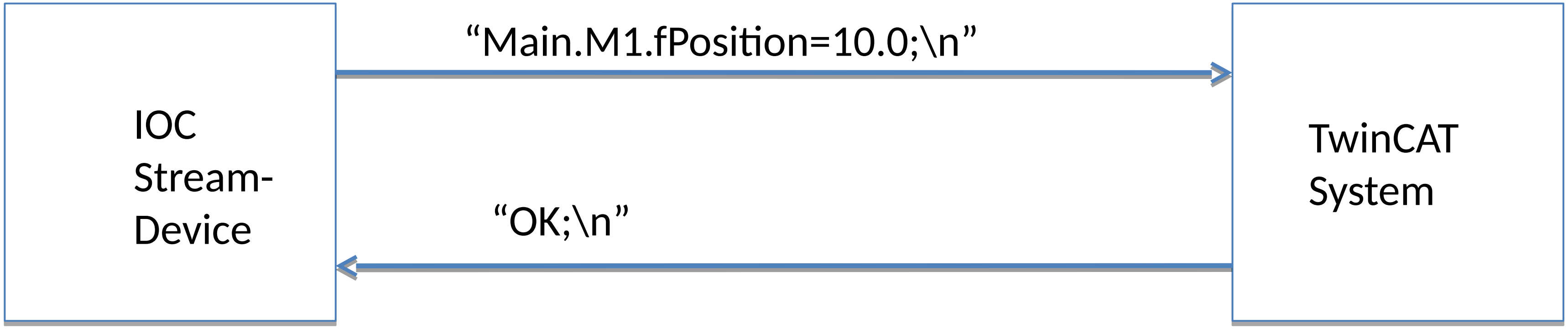


ASCII data

Low level debugging

```
$ telnet 192.168.88.48 200
Main.M1.bBusy?;\r\n
0;\r\n
```

EPICS TwinCAT



EPICS – PLC communication

- ”Move absolute” to position 200
-
  ```
  Main.M1.bExecute=0;
  Main.M1.nCommand=3;
  Main.M1.nCmdData=0;
  Main.M1.fPosition=200.0;
  Main.M1.bExecute=1
  ```
-
  ```
  (Not shown: fVelocity,fAcceleration,
              fDecelaration)
  ```
- `(Not shown: ADSPORT=851)`

EPICS from python

- Python: ”Move absolute” to position 200
- `m1 = ”LabS-ESSIIP:MC-MCU-01:m1”`
- `Position = 200`
- `caput(m1, position)`

`And when is the motor done ?`

Client EPICS MR & wait

- m1 = ”LabS-ESSIIP:MC-MCU-01:m1”
- Position = 200
- caput(m1, position, wait=true)

#And now the movement is done

Client EPICS MR & STAT field

- `#And now the movement is done`

-
  ```
  stat = caput(m1 + '.STAT')
  if (stat != 0):
      #Error handling -To be discussed
      # with ”Users”,
      # BCG, NICOS developpers
  ```

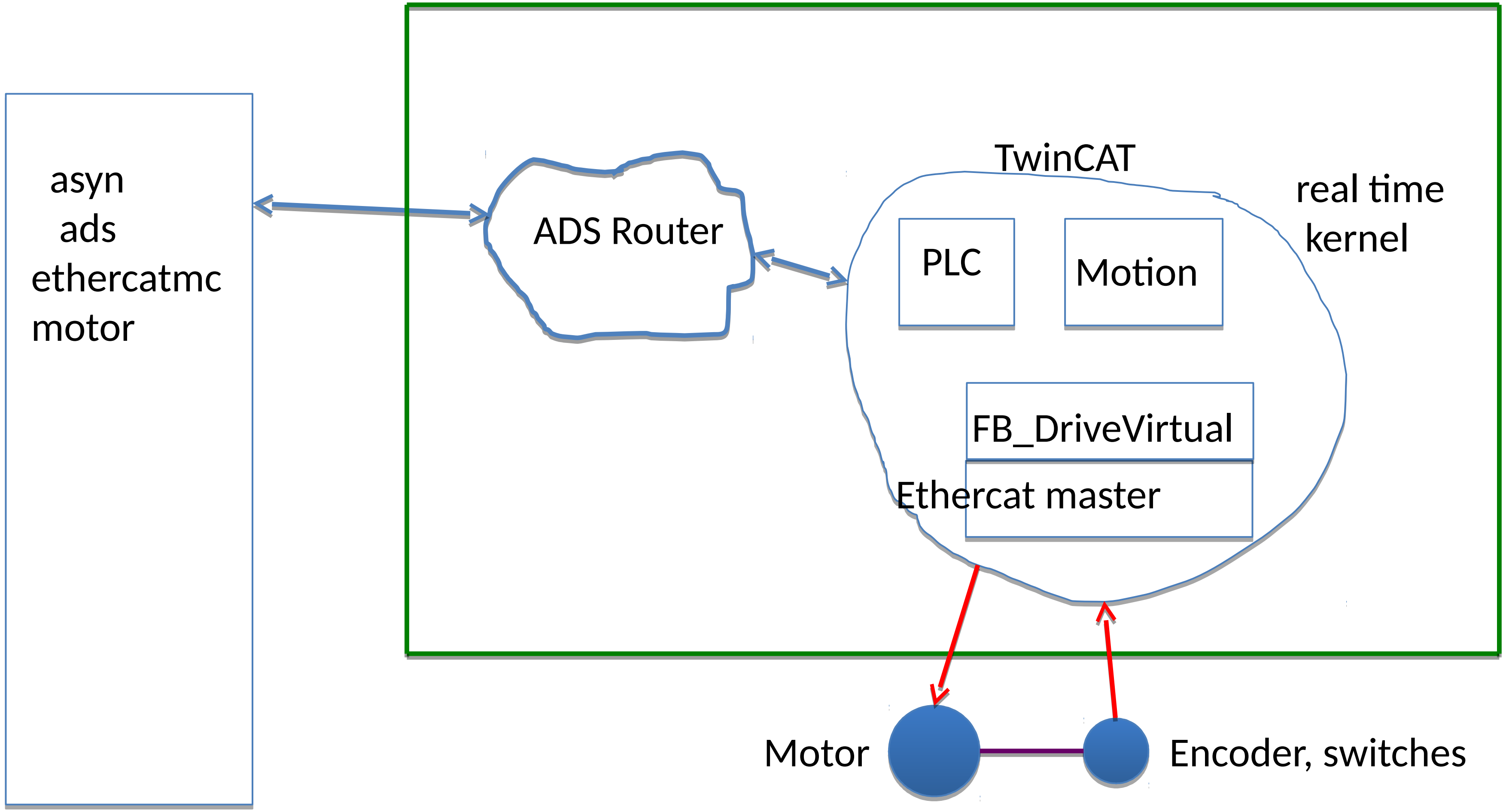
ADS

- 2015:
  Code from Beckhoff to ”talk” ADS from Windows, Linux, MacOS
- https://github.com/Beckhoff/ADS.git

- Yeah!

- https://bitbucket.org/europeanspallationsource/m-epics-twincat-ads

EPICS ads Module



That’s it

- Thanks – for listening
- Time for questions

More records

- More HW: E.g. Temperature sensor

- Add it to the PLC
- Expose the analog value in a PLC variable
- Define an ”ai” record in the IOC
- Connect record via streamdevice

Level of truth – physical axis

- When we commission a crate these values are configured:
  - Scaling/Resolution (mm/rev, steps/rev)
  - Maximum velocity
  - Default velocities (positioning, jog, homing)
  - Default acceleration, jerk
  - Soft limits
  - In target position monitor (deadband)
  - Many more

Commission MCU → motorRecord

- Read data from MCU into Record:

```
out=ADSPORT=501/.ADR.16#4001,16#16,8,5?;…
in=1.0e-1;2.0e-2;...

rdbd=0.100000, rdbd_tim=0.020000 rdbd_en=1

IOC:m1 enforceMinRetryDeadband...cfg_rdbd=0.100000
```

- Why important:
  - When is a movement ”completet” ?
  - Movements with smaller distances are ”ignored”

Extra – Record alarms

- Alarm handling

EPICS Record Alarms

- All EPICS records have a field:
  STAT
  SEVR
- STAT == 0 means ”no alarm”
- SEVR is valid when STAT != 0.
  E.g. distinguish between Major/Minor

The STAT field

- EPICS base: enum, 20 values.

```
0 == epicsAlarmNone
(read, write)
(hihi, high, lolo, low)
7  == epicsAlarmState
9  == epicsAlarmCom
14 == epicsAlarmLink
17 == epicsAlarmUDF
```

The SEVR field

- EPICS Base, enum, 4 values.

```
0 == epicsSevNone
1 == epicsSevMinor
2 == epicsSevMajor
3 == epicsSevInvalid
```

STAT + SEVR

- Combinations

```
AlarmUDF,   SevInvalid
AlarmHigh,  SevMajor/SevMinor/SevNone
AlarmLow,   SevMajor/SevMinor/SevNone
AlarmCom,   SevInvalid
AlarmState, SevMajor/SevMinor/SevNone
```

Alarms, what, when

- AlarmUDF
  (TCP/IP) connection never established
- AlarmCom:
  (TCP/IP) connection lost
- AlarmState:
  ”motor problem”
- AlarmHigh,Low:
  Soft/hard limit (See HLSV)
- AlarmState:
  MISS (See MISV)

Nicos 2017

- Suggestion: 4 States in control screen:
  OK BUSY WARN ERROR

```
if (STAT == epicsAlarmComm)
  status = ERROR
else if (STAT !=0 && SEVR ==MAJOR)
  status = ERROR
else if (!DMOV || MOVN)
  status = BUSY
else if (MISS)
  status =Warn NotReache
else if (LVIO || LLS || HLS)
  status = WARN
else
  status = OK
```