PAUL SCHERRER INSTITUT

PSI

Dirk Zimoch :: Controls Section :: Paul Scherrer Institut

Current EPICS Developments at PSI

EPICS Collaboration Meeting at ICALEPCS 2017

PAUL SCHERRER INSTITUT

PSI

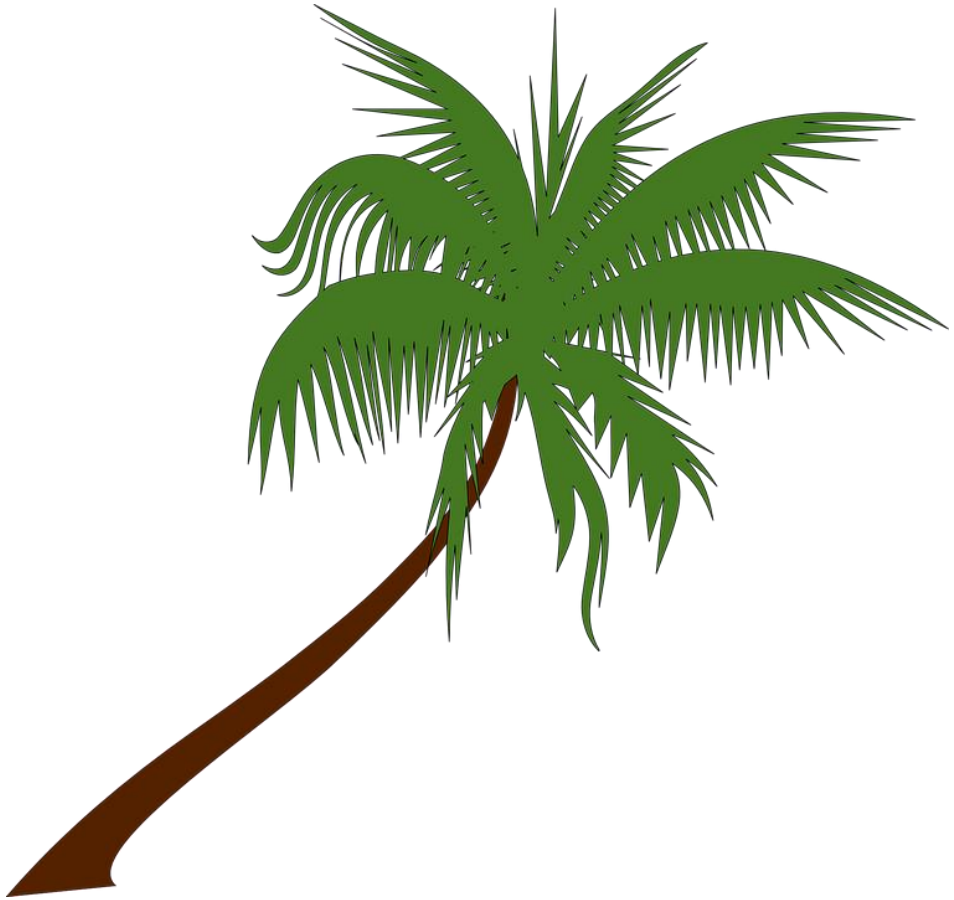
Dirk Zimoch :: Controls Section :: Paul Scherrer Institut

IOC Utilities at PSI

EPICS Collaboration Meeting at ICALEPCS 2017

Goal: Make Life Easier

- Simplify tedious work
  - Adding new scan rate
  - Adding breakpoint tables
  - Setting EPICS_CA_MAX_ARRAY_BYTES
- Help debugging with reverse lookup functions
  - Listing all aliases of a record
  - Listing all links pointing to a record
  - Listing all CA connections to a record
- Accessing info fields
  - List matching infos of all records
  - Iterator API function to traverse matching infos
  - Read infos with Channel Access



Adding New Scan Rate

- Edit menuScan.dbd the right way and rebuild

```
menu(menuScan) {
        choice(menuScanPassive,"Passive")
        choice(menuScanEvent,"Event")
        choice(menuScanI_O_Intr,"I/O Intr")
        # Periodic scans follow, ordered from slowest to fastest
        choice(menuScan10_second,"10 second")
        choice(menuScan5_second,"5 second")
        choice(menuScan2_second,"2 second")
        choice(menuScan1_second,"1 second")
        choice(menuScan_5_second,".5 second")
        choice(menuScan_3_second,".3 second")
        choice(menuScan_2_second,".2 second")
        choice(menuScan_1_second,".1 second")
}
```

Basically arbitrary name

Defines new scan rate. Up to R3.15: unit must be “second”

- Or simply execute **addScan** command in startup script:

  `addScan .3`

  – Automatically calls sysClkRateSet() on vxWorks if necessary

Adding Breakpoint Tables

- Include bpt*.dbd, edit menuConvert.dbd and then rebuild

```
menu(menuConvert) {
        choice(menuConvertNO_CONVERSION,"NO CONVERSION")
        choice(menuConvertSLOPE,"SLOPE")
        choice(menuConvertLINEAR,"LINEAR")
        choice(menuConverttypeKdegF,"typeKdegF")
        choice(menuConverttypeKdegC,"typeKdegC")
        choice(menuConverttypeJdegF,"typeJdegF")
        choice(menuConverttypeJdegC,"typeJdegC")
        choice(menuConverttypeEdegF,"typeEdegF(ixe only)")
        choice(menuConverttypeEdegC,"typeEdegC(ixe only)"
        choice(menuConvertmyConv,"myConv")
        choice(menuConvertmyOtherConv,"myOtherConv")
}
```

Basically arbitrary name

Must match name of breaktable in bpt*.dbd file

- Or simply load bpt*.dbd files and update menuConvert in startup script

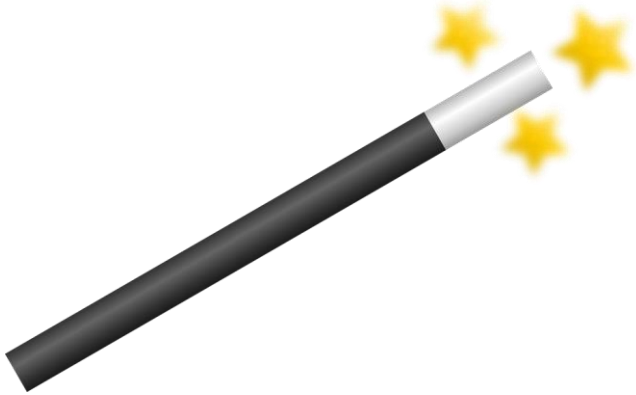
  dbLoadDatabase bptMyConv.dbd

  dbLoadDatabase bptMyOtherConv.dbd

  updateMenuConvert

Setting EPICS_CA_MAX_ARRAY_BYTES

- EPICS_CA_MAX_ARRAY_BYTES on IOC too small: array PVs inaccessible
  - (Not needed any more with R3.16)
- Check size of all array fields of all records
  - Keep in mind CA type promotion (e.g. ULONG → DOUBLE)
  - Add overhead for status, severity, time stamp, …
  - Find maximum
- Add to startup script:
  `epicsEnvSet EPICS_CA_MAX_ARRAY_BYTES,…`



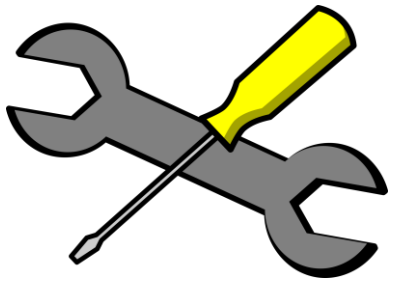
- Or simply calculate it automatically
  - Init hook checks potential array fields of all records
    - Two SynApps records needed patch for `cvt_addr()`:
      sCalcout record and table record
  - Sets EPICS_CA_MAX_ARRAY_BYTES if current value (or default) is too small

Listing All Aliases of a Record

- Translating alias name to record name is easy
  - Remote: `caget aliasname.NAME`
  - On IOC shell: `dbla aliasname`
    - Works with patterns: `dbla *XYZ`
- Finding all **aliases of a record** is difficult
  - Call `dbla` without parameter, then search the list

- Or modify `dbla` to accept **alias or record** name patterns

```
> dbla recordname
alias1 -> recordname
alias2 -> recordname
alias2 -> recordname
```



Listing All Links Pointing to a Record

- Finding link targets of a record is not too difficult
  - On IOC shell: `dbpr record,4` then search the output for links
  - Or get known link field: `dbgf record.OUT`
- Finding all links with a given **target record** is very difficult
  - Analyze record database offline and search for links

- Or use new IOC shell function: `dbll record-pattern`

```
> dbll *:X
A.DOL --> DEV1:X
B.INPA --> DEV2:X.VAL MS CP
C.OUT --> DEV3:X.RVAL PP
```

- Filter for target field

```
> dbll *:X.VAL
A.DOL --> DEV1:X
B.INPA --> DEV2:X.VAL MS CP
```

Listing All CA Connections to a Record

- It is possible to get all CA connections to an IOC
  - On IOC shell: `casr 2`
- Good luck searching the output for a **specific record**!

- Or use new IOC shell function: `cal record-pattern`

```
> cal MYRECORD
zimoch@pc1234:47643 --> MYRECORD.VAL
zimoch@pc1234:47643 --> MYRECORD.EGU
archive_user@archive_host:83542 --> MYRECORD.VAL
```

Accessing Info Fields

- Records can have info fields

```
record (ai, "REC1") {
    info(autosaveFields_pass0, "LOLO LOW LLSV LSV")
}
```

- But info fields are not easily accessible

- Use new IOC shell function: `dbli info-pattern [pattern ...]`

```
> dbli autosaveFields*
REC1.autosaveFields_pass0 "LOLO LOW LLSV LSV"
REC2.autosaveFields_pass1 "VAL"
```

Accessing Info Fields (cont.)

- New iterator API function: dbNextMatchingInfo

```
char* patternlist[] = {"autosaveFields*", NULL};
dbInitEntry(pdbbase, &dbentry);
while(dbNextMatchingInfo(&dbentry, patternlist) == 0)
{
    printf("%s.%s \"%s\"\n", dbGetRecordName(&dbentry),
    dbGetInfoName(&dbentry), dbGetInfoString(&dbentry));
}
dbFinishEntry(&dbentry);
```

NULL terminated array of pattern strings

- Goody: Use Channel Access to read info fields

```
$ caget REC1.autosaveFields_pass0
REC1.autosaveFields_pass0      LOLO LOW LLSV LSV
```

PAUL SCHERRER INSTITUT

How to get it?

- Package with utility functions (everything except Channel Access patch)
  http://epics.web.psi.ch/software/iocUtils.tgz

- Patch for Channel Access to info fields
  - for EPICS 3.15 and 3.16
    http://epics.web.psi.ch/software/caInfoFields-R3-15.patch
  - for EPICS 3.14
    http://epics.web.psi.ch/software/caInfoFields-R3-14.patch

I will try to get these changes into EPICS base.

PAUL SCHERRER INSTITUT

PSI

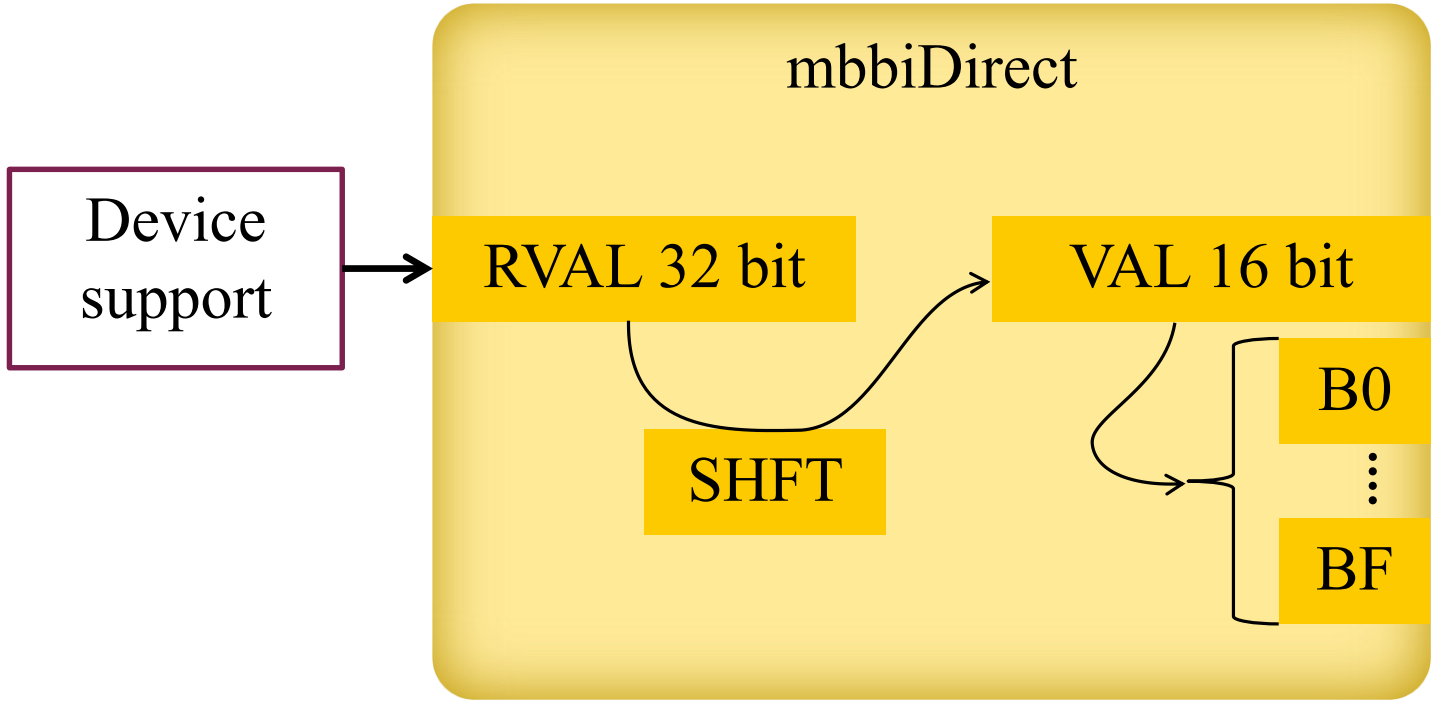
Dirk Zimoch :: Controls Section :: Paul Scherrer Institut

Extending mbbiDirect and mbboDirect Records to 32 Bit

EPICS Collaboration Meeting at ICALEPCS 2017

Current mbbiDirect

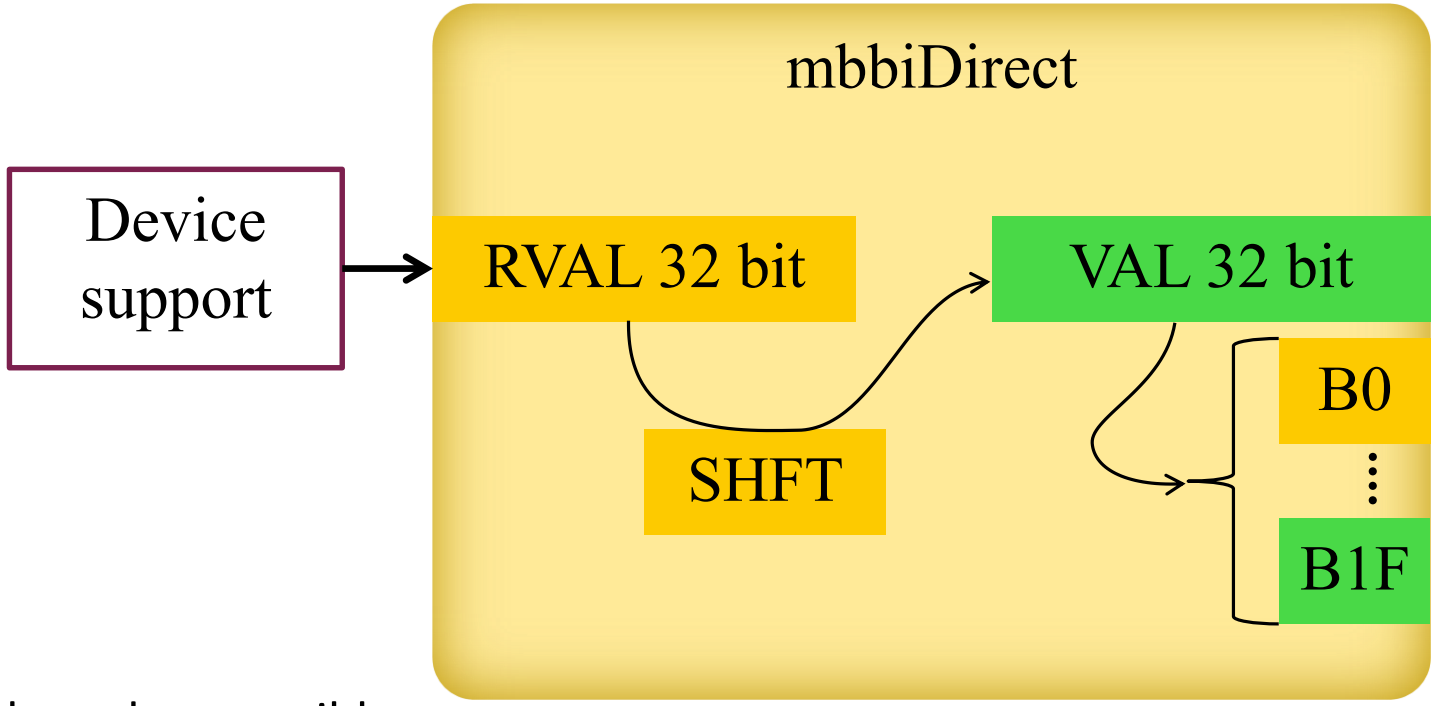
- Input (RVAL) is 32 bit
- Output (VAL, B0 … BF) is only 16 bit



- Why?

Future mbbiDirect

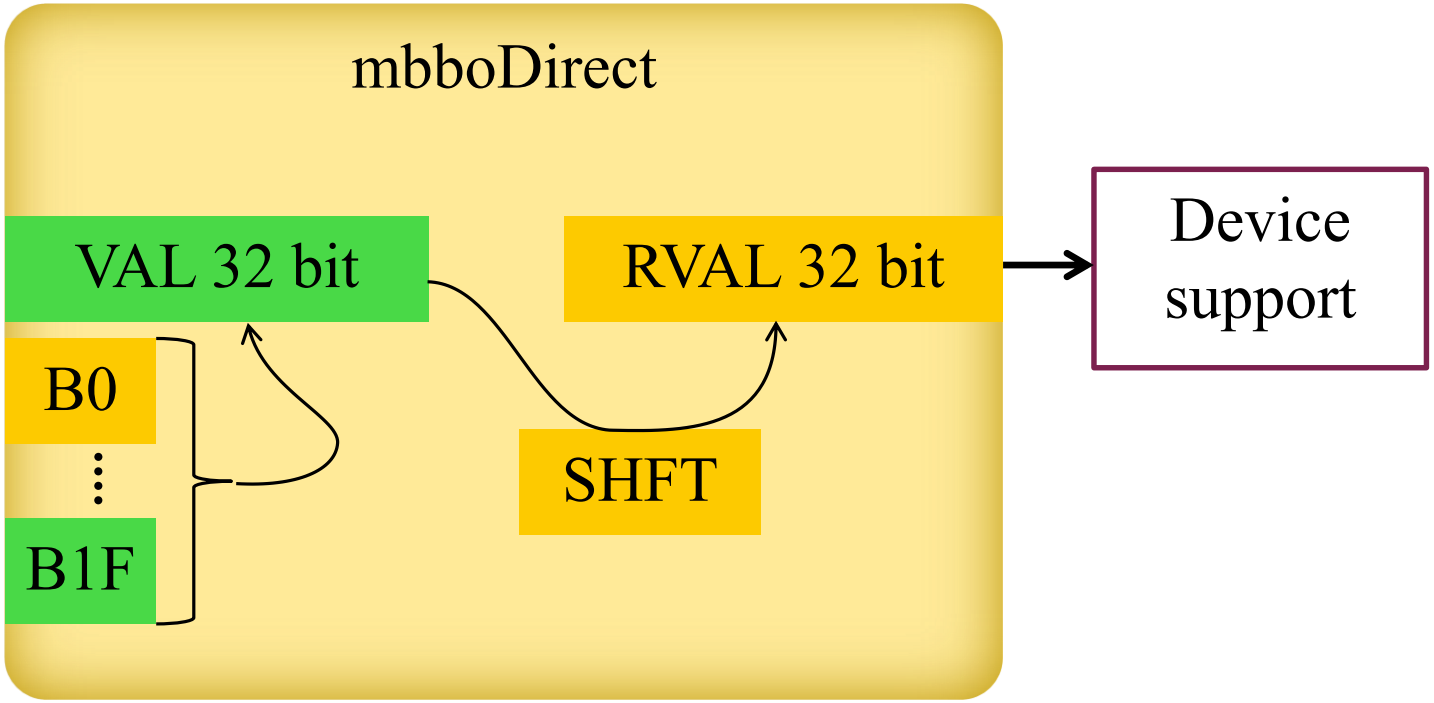
- Input (RVAL) stays 32 bit
- Output (VAL, B0 … BF, B10 … B1F) will be 32 bit as well



- Mostly backward compatible

Same for mbboDirect

- Output (RVAL) stays 32 bit
- Input (VAL, B0 … BF, B10 … B1F) will be 32 bit as well



PAUL SCHERRER INSTITUT

Compatibility

- RVAL was and will be ULONG
  - Most device supports only access RVAL and **need no change**
  - Record memory layout changes: re-compile all device supports!

- VAL was USHORT and will be LONG
  - For 16 bit values nothing will change
  - Channel Access Clients have always seen a LONG

- Device support **accessing VAL or B* fields** need modification to support higher bits.
  - Use 32 bit variables when accessing `prec->val`
  - Using pointer `&prec->val` is not safe!
  - When iterating bit fields check for macros like `mbboDirectRecordB1F`

Why not 64 bit?

- 64 bit integers are not supported by Channel Access
- **Changing RVAL to 64 bit would break all existing device support**
- Possible as new record type

What do you think?

PAUL SCHERRER INSTITUT

PSI

Dirk Zimoch :: Controls Section :: Paul Scherrer Institut

Channel Access Push
Proposal for a New Connection Method

EPICS Collaboration Meeting at ICALEPCS 2017

Channel Access Monitor Workflow

Current (simplified) work flow to set up a monitor

IOC		CA Client
Looks up PVs	←	Searches for PVs (UDP broadcast)
Search response (UDP unicast)	→	Gets host/port from response
Accepts connection	←	Connects (TCP) to host/port
Installs subscriptions	←	Set up monitors
Sends PV events	→	Processes data
Goes offline		Searches again

What's wrong?

- Some “clients” (e.g. **archiver**) flood the network with broadcasts.
  Reasons: wrong configuration, IOCs offline, …
- Archiver not a client at all, it's a **service**!

Clients vs Services

CA Client	CA Service
PVs live on IOC	
IOC sends updates	
Runs on arbitrary hosts	Runs on **well known host:port**
Can start and top at any time	Usually runs permanently
Client wants something from IOC	IOC wants something from service
IOC cannot know PVs needed by client: Configuration must be in the client	IOC knows PVs to send to service: Configuration should be in the IOC

Let the IOC **push** its PVs to the service!

- Use info fields in records for configuration
- IOC does connection management

Channel Access Push Workflow

Proposed work flow to send data to services

IOC		CA Service
Connects (TCP) to well known host:port	→	Accepts connection
Sends list of PVs (necessary?)	→	Sets up monitors
Sends PV events	→	Processes data
Goes offline		Stops monitoring

A Channel Access service would be passive

- Service needs no PV list configuration
- PV list cannot be outdated
- Service does not send search broadcasts
- IOC **pushes** PVs to service
- Event/subscription concept and code probably does not change much
- “Only” connection setup needs to be developed

What do you think?

- Does this look useful?
- Channel Access or PV Access or both?
- Anyone volunteering to implement it?