

EPICS Database Updates

Kay Kasemir

Many slides from Andrew Johnson,
APS/ANL

July 2026

Comparably New Database Features

- Wildcards for 'patch' databases
- Long Strings
- JSON Notation for links and subscriptions
- PVA links

'Patch' Database

Database files can **modify fields** of existing records or **add** new fields/records

ps.db

```
record(ai, "voltage")
{
  field(DTYP, ... INP ...
  field(SCAN, "2 second")
  field(FLNK, "current")
}

record(ai, "current")
{
  field(DTYP, ... INP ...
  field(FLNK, "power")
}

record(calc, "power")
{
  field(INPA, "voltage")
  field(INPB, "current")
  field(CALC, "A*B")
}

record(bo, "ps_enable")
...
```

limit.db

```
record(ai, "voltage")
{
  field(SCAN, ".1 second")
}

record(calc, "power")
{
  field(FLNK, "limit")
}

record(calcout, "limit")
{
  field(INPA, "power")
  field(CALC, "A < 100")
  field(OUT, "ps_enable")
}
```

st.cmd

```
dbLoadRecords "db/ps.db"

dbLoadRecords "db/limit.db"
```

Organize records.

"ps.db" can be used standalone,
or with "limit.db".

Order of loading *.db files matters!

'Patch' Database ...

Note that the record type cannot be changed.
Must use the correct record **type**, or **"*"** to patch *any* type

ps.db

```
record(ai, "voltage")
{
  field(DTYP, ... INP ...
  field(SCAN, "2 second")
  field(FLNK, "current")
}

record(ai, "current")
{
  field(DTYP, ... INP ...
  field(FLNK, "power")
}

record(calc, "power")
{
  field(INPA, "voltage")
  field(INPB, "current")
  field(CALC, "A*B")
}

record(bo, "ps_enable")
...
```

limit.db

```
record(ai, "voltage")
{
  field(SCAN, ".1 second")
}

record("*", "power")
{
  field(FLNK, "limit")
}

record(calcout, "limit")
{
  field(INPA, "power")
  field(CALC, "A < 100")
  field(OUT, "ps_enable")
}
```

In this example, the patch database doesn't care if "power" is an ai, ao, calc, calcout, ..

Error when record("*"... is unknown asserts correct order of loading *.db files.

In st.cmd, option to require "*" for overloads:

```
var dbRecordsOnceOnly 1
```

Long Strings

- Unfortunately, EPICS 'STRING' was initially defined as CHAR[40]
- Channel Access uses DBF_STRING = CHAR[40]

- Record names, INP/OUT links can now be much longer
`field(INPA, "SomeReallyLongRecordNameThatExceeds40Characters PP MS")`
..but you cannot read/modify them via channel access?!

Old Workaround

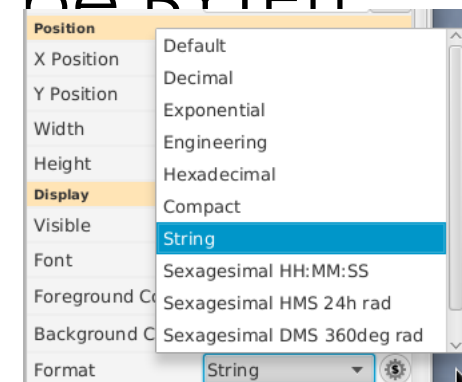
Use

```
record(waveform, "chars")
{
    field(DESC, "Character waveform for long string")
    field(FTVL, "CHAR")
    field(NELM, "200")
}
```

.. and (!) tell client that CHAR[] is not meant to be BYTE[] but long STRING.

```
caget -S .....
```

Format: String in operator interface tools



See /ics/examples/11_long_strings/strings.db and display

New Workaround

- IOC turns any string for VAL, INP, DOL into long-string CHAR[] if PV name ends in '\$'
- Long-String input and output records "Isi" and "Iso" default to VAL of type STRING, but VAL\$ becomes CHAR[]
- Clients still need to be told if CHAR[] is BYTE[] or long string!
 - caget some_Isi_record → DBR_STRING, only 40 chars
 - caget some_Isi_record.VAL\$ → DBR_CHAR[], shown as numbers
 - caget **-S** some_Isi_record.**VAL\$** → **Yay!**

JSON: “JavaScript Object Notation”

Standard text syntax to represent structured data

```
{  
  number: 3.14,  
  text: "Hello!",  
  substructure: {  
    numbers: [1, 2, 3, 4]  
  }  
}
```

EPICS starts to use it to provide details for links

JSON Notation for INP links

- New support for JSON notation allows providing more detail
- First practical use: {const:...}

```
record(lsi, "lsi_init")
{
  field(SIZV, "500")
  field(INP, {const:"Hello, this is a string and it can be really long!"})
  field(PINI, "YES")
}
```

```
record(waveform, "array")
{
  field(FTVL, "CHAR")
  field(NELM, "200")
  field(INP, {const:[65, 66, 67, 68, 69]})
  field(PINI, "YES")
}
```

JSON Notation for Subscription Filters

- In one window, run

```
cd /ics/examples/01_first_steps  
softIoc -m S=demo -d first.db
```

- In other window, compare the following

```
camonitor demo:limit  
camonitor 'demo:limit.{ts:{}}'
```

(run several times...)

```
camonitor demo:ramp  
camonitor 'demo:ramp.{dbnd:{abs:1.9}}'
```

JSON Notation for Network links

- Database Link

```
field(INP, "SomeRecordInSameIOC")
```

- Network = Channel Access Link

```
field(INP, "SomeRecordInSameIOC CA")
```

```
field(INP, "SomeRecordInOtherIOC")
```

- Network = PV Access Link

```
field(INP, {pva:{pv:"SomeRecord"}})
```

For the longest time, "Network Links" used the only protocol available, Channel Access.

Now, there is a way to force PVA Links.

Eventually, there should be an IOC option like

```
EPICS_NETWORK_LINK_DEFAULT=CA/PVA
```

More likely to come

- Extensible links

<https://epics.anl.gov/base/R7-0/8-docs/links.html>

- Channel filters

<https://epics.anl.gov/base/R7-0/6-docs/filters.html>