COSYLAB
CONTROL SYSTEM LABORATORY

EPICS

EPICS Lecture @ KEK

Database Concepts

Takashi Nakamoto
June 25th, 2013

Based on presentation by A. Johnson, APS

Outline

- Records
- Fields and field types
- Record Scanning
- Input and Output record types
- Links, link address types
- Connecting records together
- Protection mechanisms
- Alarms, deadbands, simulation and security

Database = Records + Fields + Links

- **A control system using EPICS will contain one or more IOCs**
- **Each IOC loads one or more Databases telling it what to do**
- **A Database is a collection of Records of various types**
- **A Record is an object with:**
  - A unique name
  - A behaviour defined by its record type (class)
  - Controllable properties (fields)
  - Optional associated hardware I/O (device support)
  - Links to other records

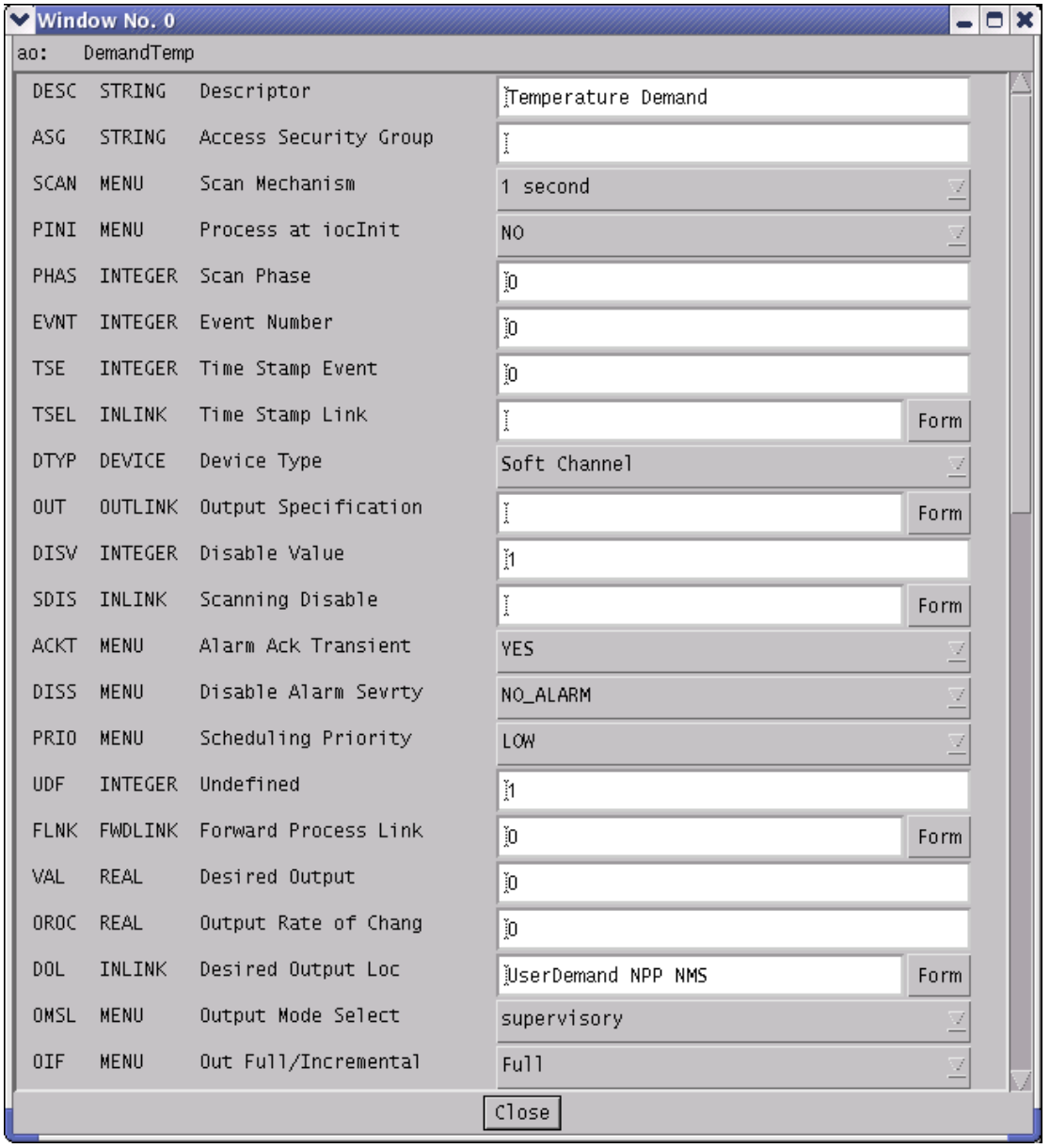
Record Activity

- **Records are active — they can do things:**
  - Get data from other records or from hardware
  - Perform calculations
  - Check values are in range & raise alarms
  - Put data to other records or to hardware
  - Activate or disable other records
  - Wait for hardware signals (interrupts)
- **What a record does depends upon its record type and the settings of its fields**
- **No action occurs unless a record is processed**

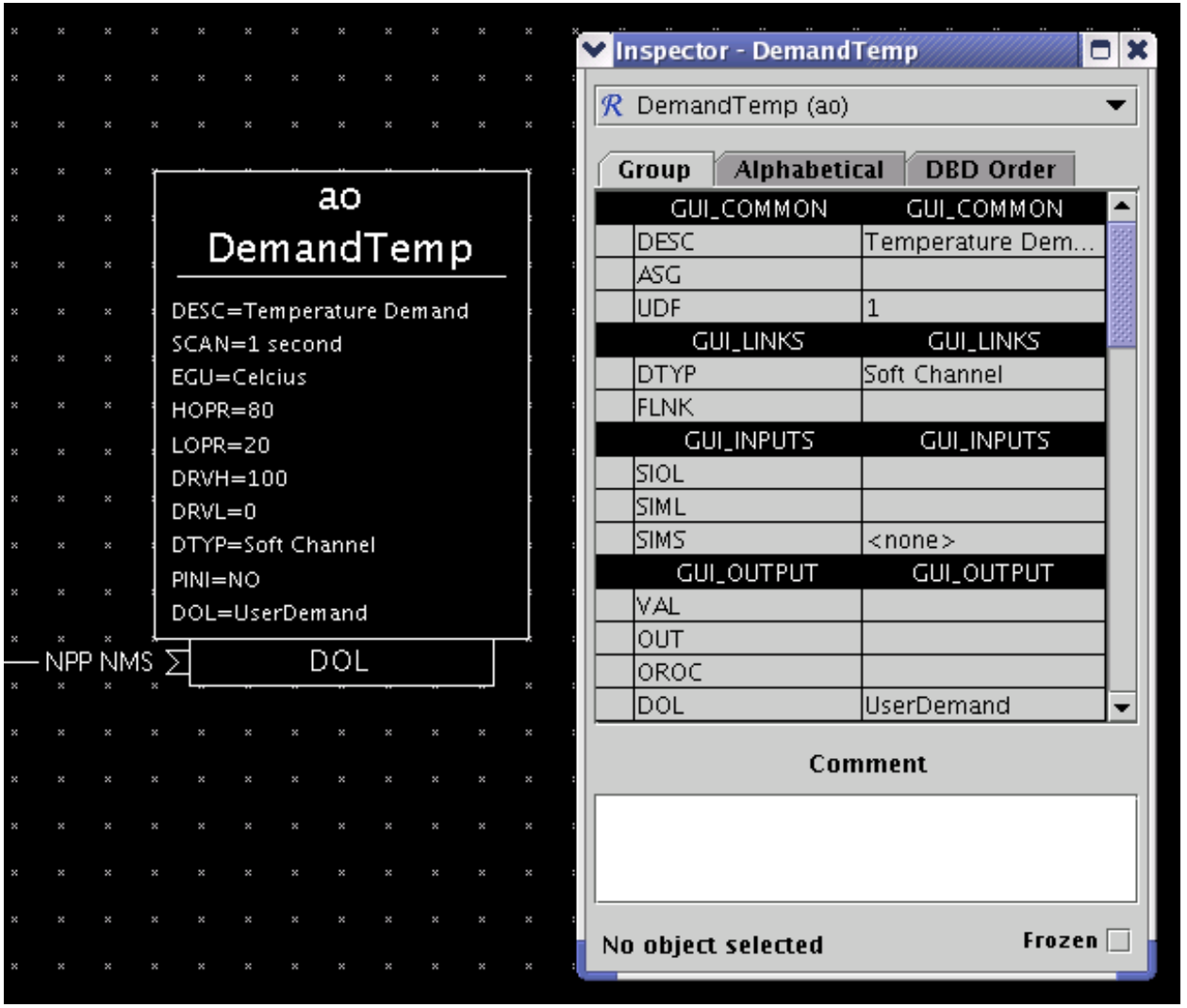
How is a Record implemented?

- **A ‘C’ structure with both data storage and pointers to record type information**
- **A record definition within a database provides**
  - Record name
  - The record’s type
  - Values for each design field
- **A record type provides**
  - Definitions of all the fields
  - Code which implements the record behaviour
- **New record types can be added to an application as needed**

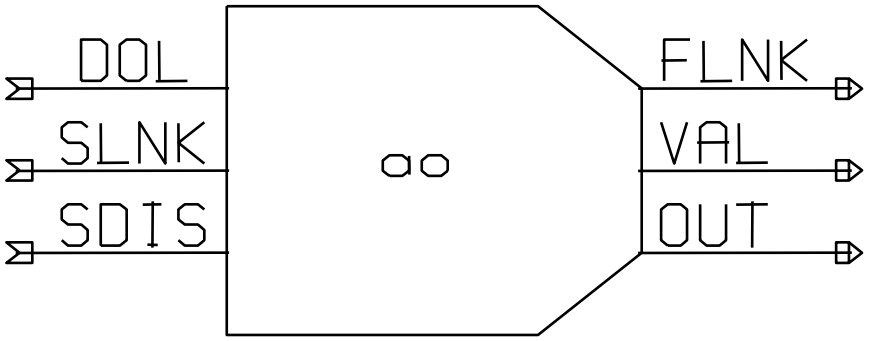
One view of a Record



A graphical view of a Record



Another graphical view of a Record



The small CapFast symbol for an Analogue Output record

The IOC's view

The full **.db** file entry for an Analogue Output Record

```
record(ao,"DemandTemp") {
  field(DESC,"Temperature")
  field(ASG,"")
  field(SCAN,"Passive")
  field(PINI,"NO")
  field(PHAS,"0")
  field(EVNT,"0")
  field(DTYP,"VMIC 4100")
  field(DISV,"1")
  field(SDIS,"")
  field(DISS,"NO_ALARM")
  field(PRIO,"LOW")
  field(FLNK,"")
  field(OUT,"#C0 S0")
  field(OROC,"0.0e+00")
  field(DOL,"")
  field(OMSL,"supervisory")
  field(OIF,"Full")
  field(PREC,"1")
  field(LINR,"NO
    CONVERSION")
  field(EGUF,"100")
  field(EGUL,"0")
  field(EGU,"Celcius")
  field(DRVH,"100")
  field(DRVL,"0")
  field(HOPR,"80")
  field(LOPR,"10")
  field(HIHI,"0.0e+00")
  field(LOLO,"0.0e+00")
  field(HIGH,"0.0e+00")
  field(LOW,"0.0e+00")
  field(HHSV,"NO_ALARM")
  field(LLSV,"NO_ALARM")
  field(HSV,"NO_ALARM")
  field(LSV,"NO_ALARM")
  field(HYST,"0.0e+00")
  field(ADEL,"0.0e+00")
  field(MDEL,"0.0e+00")
  field(SIOL,"")
  field(SIML,"")
  field(SIMS,"NO_ALARM")
  field(IVOA,"Continue
    normally")
  field(IVOV,"0.0e+00")
}
```

This shows only the design fields, there are other fields which are used only at run-time

Fields are for...

- **Defining**
  - What causes a record to process
  - Where to get/put data from/to
  - How to turn raw I/O data into a numeric engineering value
  - Limits indicating when to report an alarm
  - When to notify value changes to a client monitoring the record
  - A Processing algorithm
  - Anything else which needs to be set for each record of a given type
- **Holding run-time data**
  - Input or output values
  - Alarm status, severity and acknowledgements
  - Processing timestamp
  - Other data for internal use

Field types

- *Fields can contain*
  - ***Integers***
    - **char, short or long**
    - **signed or unsigned**
  - ***Floating-point numbers***
    - **float or double**
  - ***Strings***
    - **maximum useful length is 40 characters**
  - ***Menu choices***
    - **select one from up to 16 strings**
    - **stored as a short integer**
  - ***Links***
    - **to other records in this or other IOCs**
    - **to hardware signals (device support)**
    - **provide a means of getting or putting a value**
  - ***Other private data***
    - **not directly accessible**

All Records have these fields

Design fields

`NAME`	*60 Character unique name (using more than 28 characters can cause problems)*
`DESC`	*28 Character description*
`ASG`	*Access security group*
`SCAN`	*Scan mechanism*
`PHAS`	*Scan order (phase)*
`PINI`	*Process at IOC initialization?*
`PRIO`	*Scheduling priority*
`SDIS`	*Scan disable input link*
`DISV`	*Scan disable value*
`DISS`	*Disabled severity*
`FLNK`	*Forward link*

Run-time fields

`PROC`	*Force processing*
`PACT`	*Process active*
`STAT`	*Alarm status*
`SEVR`	*Alarm severity*
`TPRO`	*Trace processing*
`UDF`	*Set if record value undefined*
`TIME`	*Time when last processed*

Record Scanning

- **`SCAN` field is a menu choice from**
  - Periodic — 0.1 seconds .. 10 seconds
  - I/O Interrupt (if device supports this)
  - Soft event — `EVNT` field
  - Passive (default)
- **The number in the `PHAS` field allows processing order to be set within a scan**
  - Records with `PHAS=0` are processed first
  - Then those with `PHAS=1` , `PHAS=2` etc.
- **Records with `PINI=YES` are processed once at startup**
- **`PRIO` field selects Low/Medium/High priority for Soft event and I/O Interrupts**
- **A record is also processed whenever any value is written to its `PROC` field**

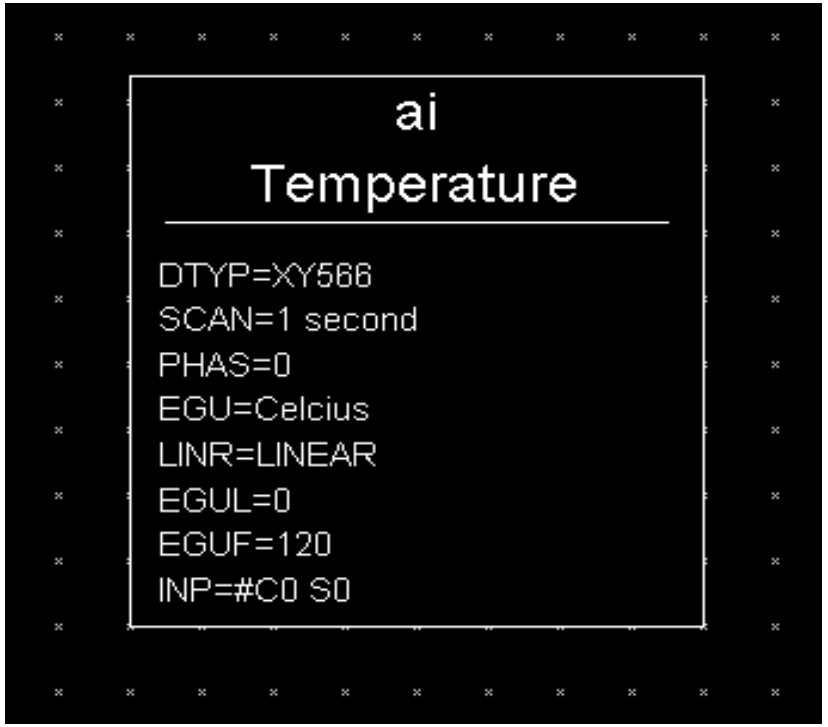
Input records often have these fields

INP	***Input link***
DTYP	***Device type***
RVAL	***Raw data value***
VAL	***Engineering value***
LOPR	***Low operator range***
HOPR	***High operator range***

Analogue I/O records have these fields

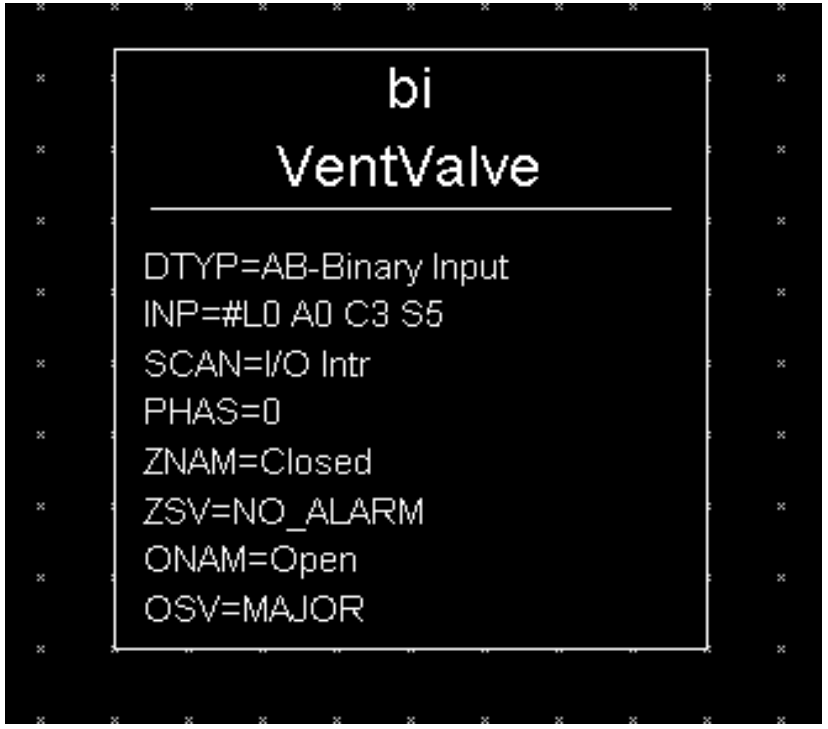
EGU	***Engineering unit string***
LINR	***Unit conversion control:*** No conversion, Linear, Slope, *breakpoint table name*
EGUL	***Low engineering value***
EGUF	***High engineering value***
ESLO	***Unit conversion slope***
EOFF	***Unit conversion offset***

Periodicly scanned Analog Input



- **Analogue Input “Temperature”**
- **Reads from the Xycom XY566 ADC Card 0 Signal 0**
- **Gets a new value every second**
- **Data is converted from ADC range to 0..120 Celsius**

Interrupt scanned binary Input



- **Binary Input “VentValve”**
- **Reads from Allen-Bradley TTL I/O Link 0, Adaptor 0, Card 3, Signal 5**
- **Processed whenever value changes**
- **0 = “Closed”, 1 = “Open”**
- **Major alarm when valve open**

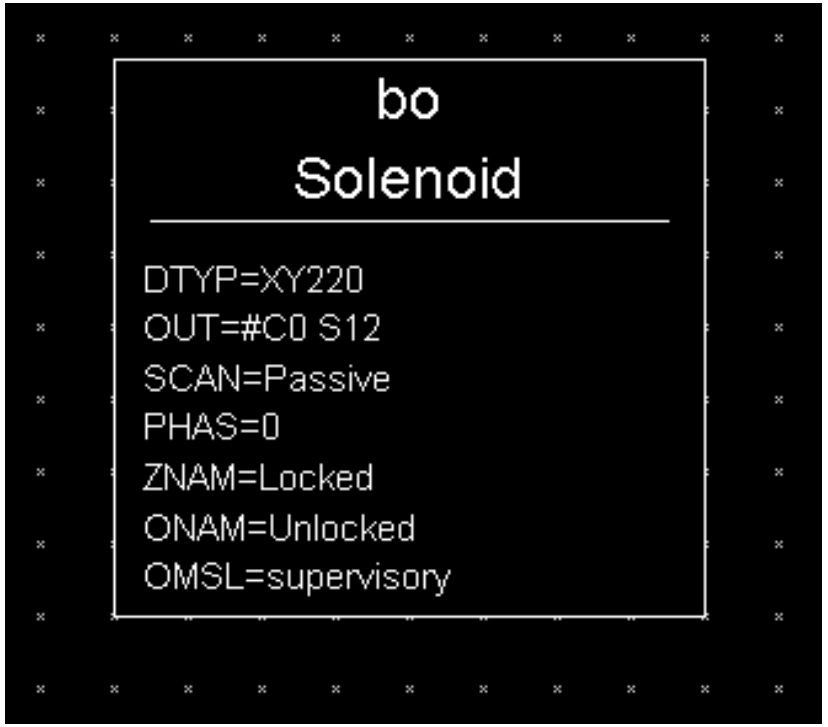
Output records often have these fields

OUT	***Output link***
DTYP	***Device type***
VAL	***Engineering value***
RVAL	***Raw output value***
DOL	***Input link to fetch output value***
OMSL	***Output mode select:*** Supervisory, Closed Loop
LOPR	***Low operator range***
HOPR	***High operator range***

Output records often have these fields

OROC	***Output rate of change***
OIF	***Incremental or Full output***
OVAL	***Output value***
DRVH	***Drive high limit***
DRVL	***Drive low limit***
IVOA	***Invalid output action***
IVOV	***Invalid output value***
RBV	***Read-back value***

Passive Binary output



- **Binary Output “Solenoid”**
- **Controls Xycom XY220 Digital output Card 2 Signal 12**
- **Record is only processed by**
  - Channel Access ‘put’ to a `PP` field (e.g. `.VAL`)
  - Another record writes to a `PP` field
  - Forward Link from another record
  - Another record reads this with `PP`

Links

A link is a type of field, and is one of

- **Input link**
  - Fetches data
- **Output link**
  - Writes data
- **Forward link**
  - Points to the record to be processed once this record finishes processing

Input and Output links may be...

- **Constant numeric value, eg:**

  0

  3.1415926536

  1.6e-19

- **Hardware link**

  A hardware I/O signal selector, the format of which depends on the device support layer

- **Process Variable link — the name of a record, which at run-time is resolved into**
  - Database link

    Named record is in this IOC
  - Channel Access link

    Named record not found in this IOC

Hardware links

VME_IO	*#Cn Sn @parm*	
		Card, Signal
INST_IO	*@parm*	
CAMAC_IO	*#Bn Cn Nn An Fn @parm*	
		Branch, Crate, Node, Address, Function
AB_IO	*#Ln An Cn Sn @parm*	
or	*#Ln Pn Cn Sn Fn @parm*	
		Link, Adaptor, Card, Signal, Flag
GPIB_IO	*#Ln An @parm*	
		Link, Address
BITBUS_IO	*#Ln Nn Pn Sn @parm*	
		Link, Node, Port, Signal
BBGPIB_IO	*#Ln Bn Gn @parm*	
		Link, Bitbus Address, GPIB Address
VXI_IO	*#Vn Cn Sn @parm*	
or	*#Vn Sn @parm*	
		Frame, Slot, Signal

Database links

These comprise:

- ***The name of a record in this IOC***
  `myDb:myRecord`
- ***An optional field name***
  `.VAL` (default)
- ***Process Passive flag***
  `NPP` (default)
  `PP`
- ***Maximize Severity flag***
  `NMS` (default)
  `MS`

For example:

`M1:current.RBV NPP MS`

- *NB: An input database link with* **`PP`** *set that is pointing to an asynchronous input record will not wait for the new value from that record*

Channel Access links

- *Specified like a database link*
- *Name specifies a record not found in this IOC*
- *Use Channel Access protocol to communicate with remote IOC*
- *May include a field name (default* **.VAL***)*
- **PP** *Link flags are ignored:*
  - *Input links are always* **NPP**
  - *Output links follow* **PP** *attribute of destination field*
  - *This behaviour is identical to all other CA clients*
- **MS** *Link flags apply to Input links:*
  - *Input links honour a given* **NMS** *(default) or* **MS** *flag*
  - *Output links are always* **NMS**
- *Additional flags for CA links*

  CA ***Forces a "local" link to use CA***

  CP ***On input link, process this record on CA monitor event***

  CPP ***Like* CP *but only process if SCAN is Process Passive***

Link flag summary

Type	Input Links	Output Links
DB	.PP or **.NPP** .MS or **.NMS**	.PP or **.NPP** .MS or **.NMS**
CA	Always **.NPP** .MS or **.NMS** .CA to force link type. .CP to process this record on change. .CPP is like .CP but only process if SCAN=Passive	.PP behavior of destination field. Always **.NMS** .CA to force link type.

Chapter 5 of the IOC Application Developer's Guide covers record links and scanning in detail, and is worth reading.

Device Support

- Records do not access hardware directly
- The Device Support layer performs I/O operations on request
- A particular device support provides I/O for a single record type
- The `DTYP` field determines which device support to use
- The device support selected determines the format of the link (`INP` or `OUT` field) containing device address information
- Adding new device support does not require change to the record software
- Device support may call other software to do work for it (Driver Support)

Synchronous vs Asynchronous I/O

- **EPICS rules do not allow device support to busy-wait (delay record processing while waiting for the results of a slow I/O operation)**
  - Fast I/O can be handled synchronously
  - Slow operations must operate asynchronously
- **Register-based VME cards usually give an immediate response: synchronous**
- **When called, synchronous device support performs all I/O before returning**
- **Serial and most I/O field-bus devices take a long time (>10ms) to return data: asynchronous**
- **Asynchronous device support starts I/O when record calls it, flags it as incomplete by setting `PACT` true before returning**
- **Once results are available (CPU interrupt), device support calls the record's process routine which finishes the operation**

Soft Device Support

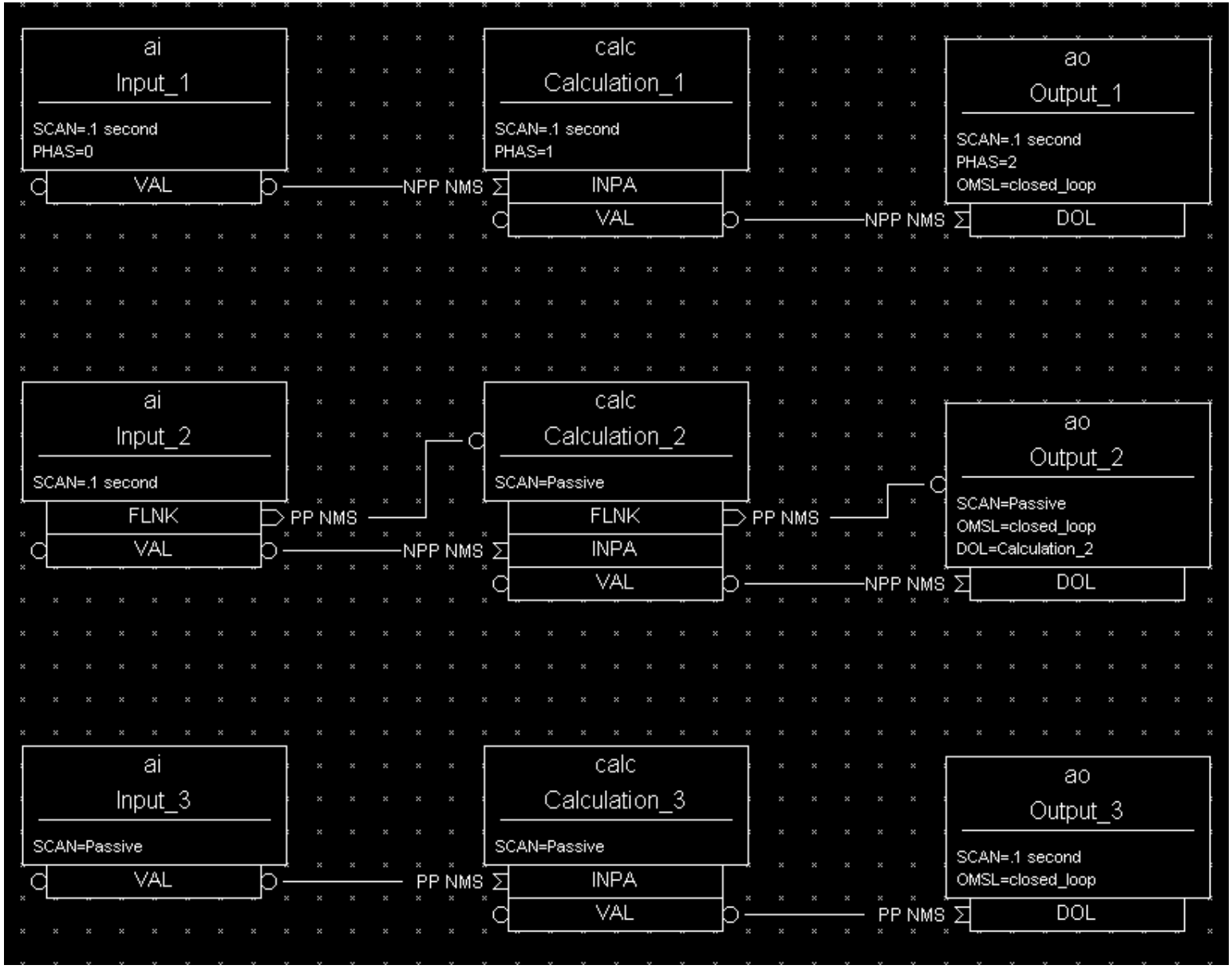
- **“Hard” Input and Output records perform hardware I/O via device support**
- **“Soft” records access data from other records via DB or CA links**
- **2 or 3 kinds of support are provided in recent R3.14 releases:**
  - Soft Channel
    - *Get/Put* `VAL` *through link, no units conversion preformed*
  - Async Soft Channel (new, for output records only)
    - *Put* `VAL` *through CA link, no conversions, wait for completion*
  - Raw Soft Channel
    - *Inputs*
      - Get `RVAL` via input link
      - Convert `RVAL` to `VAL` (record-type specific)
    - *Outputs*
      - Convert `VAL` to `RVAL` (record-type specific)
      - Put `RVAL` to output link

Forward links

- *Usually a Database link, referring to a record in same IOC*
- *Forward linking via Channel Access is possible, must explicitly name the* `PROC` *field of the remote record*
- *No flags (*`PP`*,* `NMS` *etc.)*
- *Destination record is only processed if it has*

  `SCAN = Passive`
- *Does not pass a value, just causes subsequent processing*

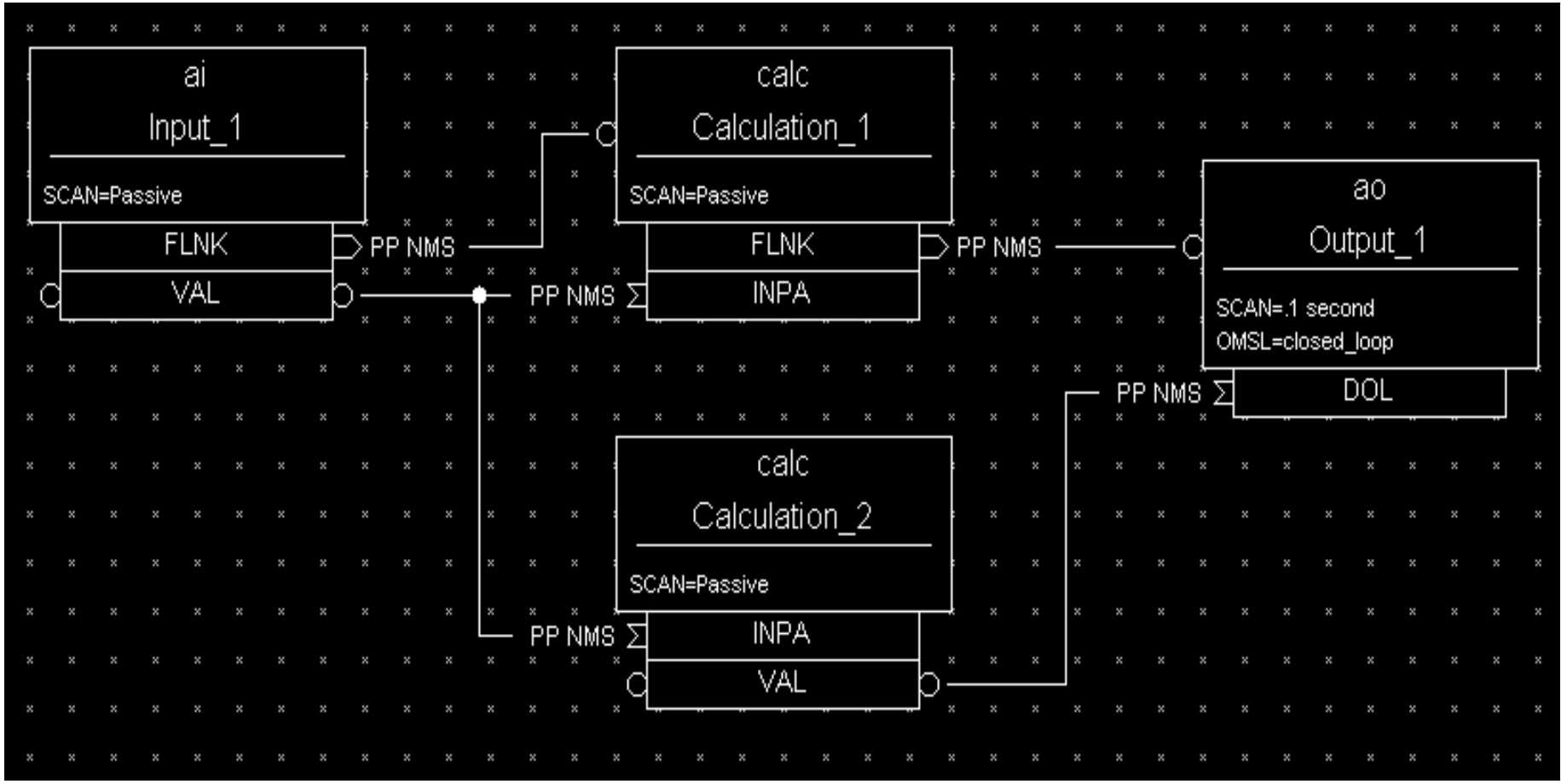
Processing chains



The PACT field

- **Every record has a boolean run-time field called `PACT` (Process Active)**
- **`PACT` breaks loops of linked records**
- **It is set to 'true' early in the act of processing the record**
  - `PACT` is true whenever a link in that record is used to get/put a value
- **`PACT` is set to false after record I/O and forward link processing are finished**
- **A `PP` link can never make a record process if it has `PACT` true**
  - Input links take the current value
  - Output links just put their value

What happens here?



Preventing records from processing

- It is useful to be able to stop an individual record from processing on some condition
- Before record-specific processing is called, a value is read through the `SDIS` input link into `DISA`
- If `DISA=DISV`, the record will not be processed
- A disabled record may be put into an alarm by giving the desired severity in the `DISS` field
- The `FLNK` of a disabled record is never triggered

How are records given CPU time?

Several IOC tasks are used:

- **callback (3 priorities) — I/O Interrupt**
- **scanEvent — Soft Event**
- **scanPeriod — Periodic**
  - A separate task is used for each scan period
  - Faster scan rates are given a higher task priority (if supported by the IOC's Operating System)
- **Channel Access tasks use lower priority than record processing**
  - If a CPU spends all its time doing I/O and record processing, you may be unable to control or monitor the IOC via the network

Lock-sets

- **Prevent a record from being processed simultaneously from two scan tasks**
- **A lock-set is a group of records interconnected by database links:**
  - Output links
  - Forward links
  - Input links which are `PP` or `MS`
  - Any link transporting an Array
- **Lock-sets are determined automatically by the IOC at start-up, or whenever a database link is added, deleted or modified**

You can split a lock set with

- **Channel Access links, using `CA` flag**
- **Database links which are both `NPP` and `NMS`**

Alarms

- **Every record has the fields**

  `SEVR` Alarm Severity

  NONE, MINOR, MAJOR, INVALID

  `STAT` Alarm Status (reason)

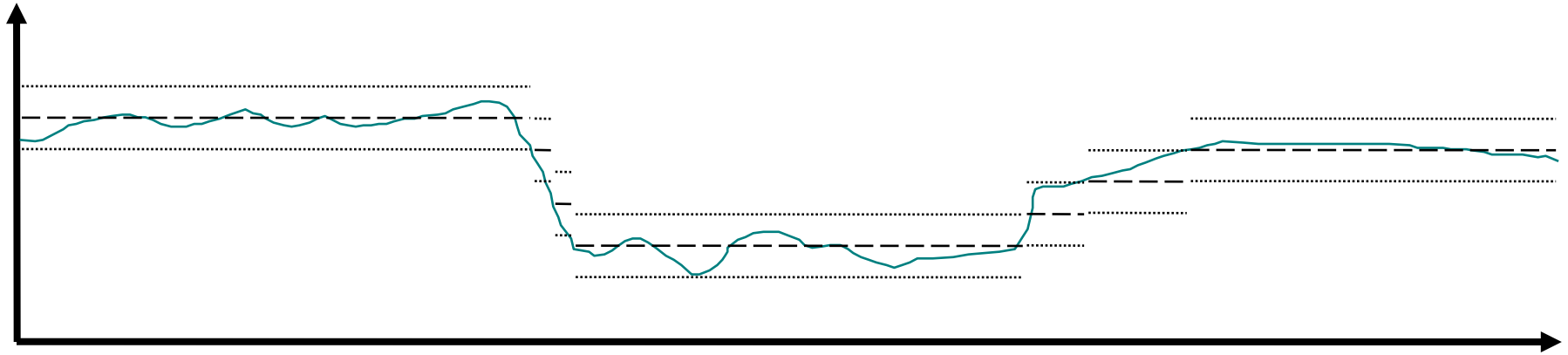
  READ, WRITE, UDF, HIGH, LOW, STATE, COS, CALC, DISABLE, etc.

- **Most numeric records check `VAL` against `HIHI`, `HIGH`, `LOW` and `LOLO` fields after the value has been determined**
- **The `HYST` field prevents alarm chattering**
- **A separate severity can be set for each numeric limit (`HHSV`, `HSV`, `LSV`, `LLSV`)**
- **Discrete (binary) records can raise alarms on entering a particular state, or on a change of state (COS)**

Change notification: Monitor deadbands

Channel Access notifies clients which are monitoring a numeric record when

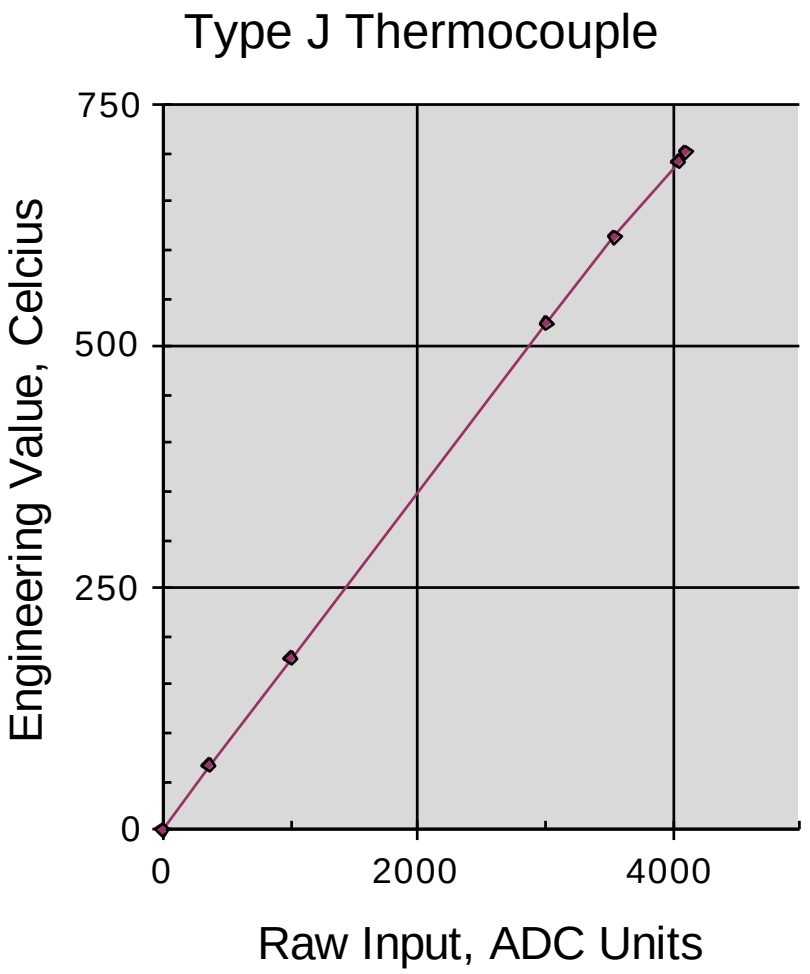
- `VAL` changes by more than the value in field:
  - `MDEL` Value monitors
  - `ADEL` Archive monitors
- Record's Alarm Status changes
  - `HYST` Alarm hysteresis
- Analogue Input record provides smoothing filter to reduce input noise (`SMOO`)



Breakpoint Tables

- Analogue Input and Output records can do non-linear conversions from/to the raw hardware value
- Breakpoint tables interpolate values from a given table
- To use, set the record's LINR field to the name of the breakpoint table you want to use
- Example breakpoint table (in some loaded .dbd file)

```
breaktable(typeKdegC) {
        0.000000   0.000000
      299.268700  74.000000
      660.752744 163.000000
     1104.793671 274.000000
     1702.338802 418.000000
     2902.787322 703.000000
     3427.599045 831.000000
     ...
}
```



Simulation

- **Input and output record types often allow simulation of hardware interfaces**
  - `SIML` Simulation mode link
  - `SIMM` Simulation mode value
  - `SIOL` Simulation input link
  - `SIMS` Simulation alarm severity
- **Before using its device support, a record reads `SIMM` through the `SIML` link**
- **If `SIMM=YES`, device support is ignored; record I/O uses the `SIOL` link instead**
- **An alarm severity can be set whenever simulating, given by `SIMS` field**

Access Security

- **A networked control system must have the ability to enforce security rules**
  - Who can do what from where, and when?
- **In EPICS, security is enforced by the CA server (typically the IOC).**
- **A record is placed in the Access Security Group named in its `ASG` field**
  - DEFAULT is used if no group name is given
- **Rules for each group determine whether a CA client can read or write to records in the group, based on**
  - Client user ID
  - Client IP address
  - Access Security Level of the field addressed
  - Values read from the database

Access Security Configuration File

- Security rules are loaded from an Access Security Configuration File, for example:

```
UAG(users) {user1, user2}
HAG(hosts) {host1, host2}
ASG(DEFAULT) {
    RULE(1, READ)
    RULE(1, WRITE) {
        UAG(users)
        HAG(hosts)
    }
}
```

- If no security file is loaded, Security will be turned off and nothing refused
- For more details and the rule syntax, see Chapter 8 of the IOC Application Developers Guide