

What is an EPICS Database?

Andrew Johnson
October 26, 2004

Argonne National Laboratory

**Advanced
Photon
Source**



Office of Science
U.S. Department of Energy

*A U.S. Department of Energy
Office of Science Laboratory
Operated by The University of Chicago*





- ◆ **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





- ◆ **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

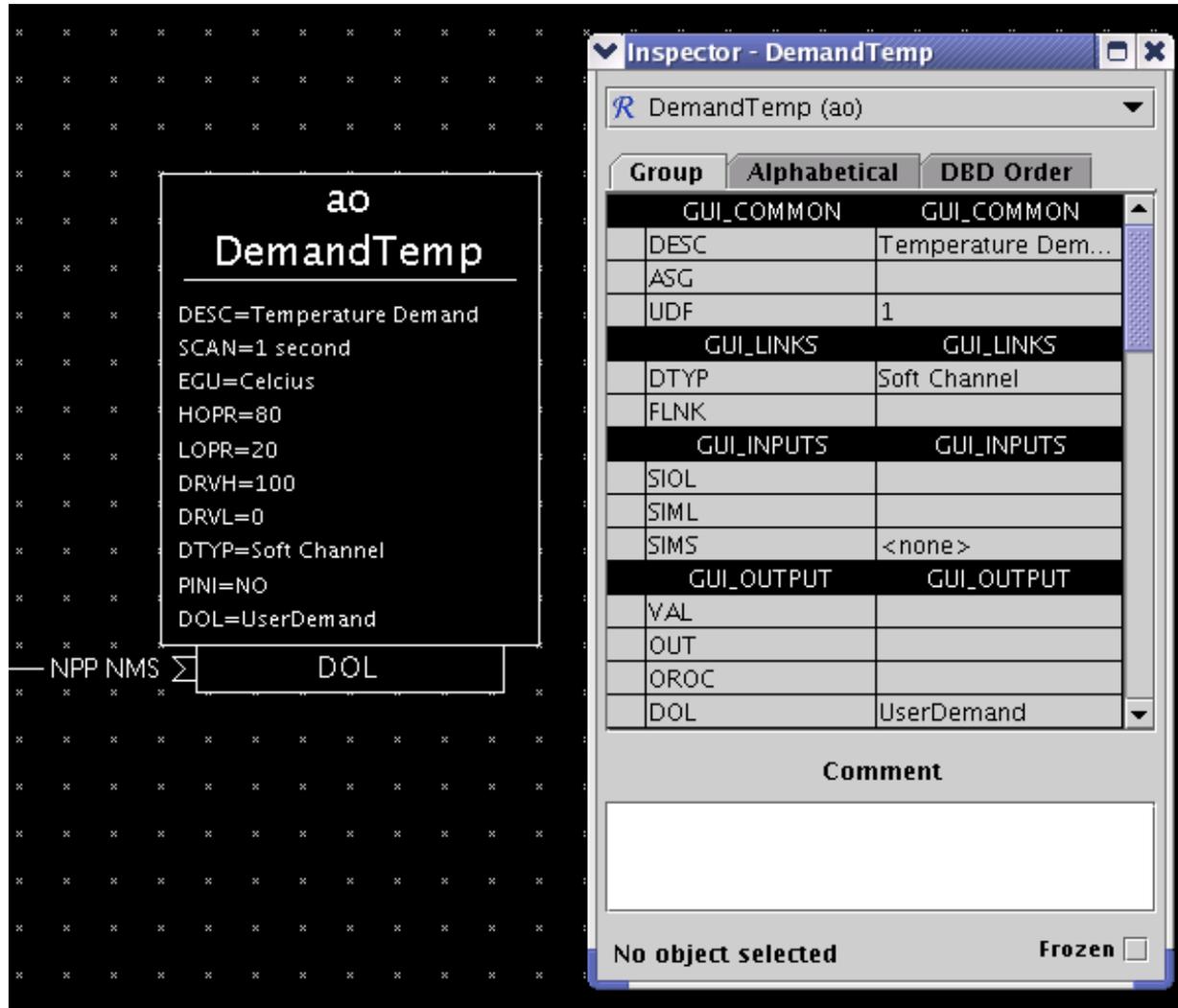
Window No. 0

ao: DemandTemp

DESC	STRING	Descriptor	Temperature Demand
ASG	STRING	Access Security Group	
SCAN	MENU	Scan Mechanism	1 second
PINI	MENU	Process at iocInit	NO
PHAS	INTEGER	Scan Phase	0
EVNT	INTEGER	Event Number	0
TSE	INTEGER	Time Stamp Event	0
TSEL	INLINK	Time Stamp Link	Form
DTYP	DEVICE	Device Type	Soft Channel
OUT	OUTLINK	Output Specification	Form
DISV	INTEGER	Disable Value	1
SDIS	INLINK	Scanning Disable	Form
ACKT	MENU	Alarm Ack Transient	YES
DISS	MENU	Disable Alarm Sevrty	NO_ALARM
PRIO	MENU	Scheduling Priority	LOW
UDF	INTEGER	Undefined	1
FLNK	FWDLINK	Forward Process Link	Form
VAL	REAL	Desired Output	0
OROC	REAL	Output Rate of Chang	0
DOL	INLINK	Desired Output Loc	UserDemand NPP NMS
OMSL	MENU	Output Mode Select	supervisory
OIF	MENU	Out Full/Incremental	Full

Close

A graphical view of a Record



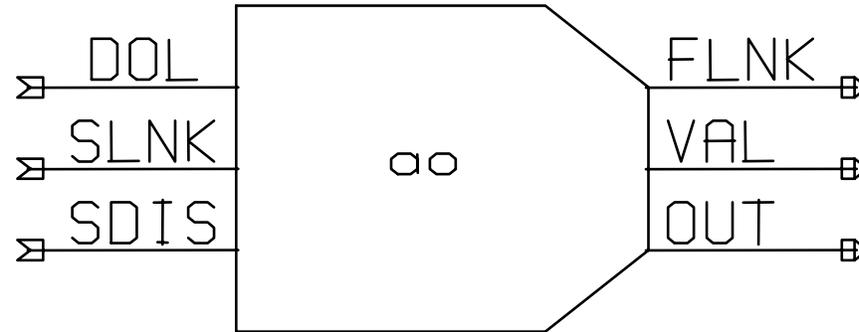
The image shows a graphical user interface for a record. On the left, a black panel displays the record name 'ao DemandTemp' and its parameters: DESC=Temperature Demand, SCAN=1 second, EGU=Celcius, HOPR=80, LOPR=20, DRVH=100, DRVL=0, DTYP=Soft Channel, PINI=NO, and DOL=UserDemand. A box labeled 'DOL' is highlighted, with 'NPP NMS' and a summation symbol nearby. On the right, an 'Inspector - DemandTemp' window shows a table of parameters grouped by category: GUI_COMMON, GUI_LINKS, GUI_INPUTS, and GUI_OUTPUT. The 'DOL' parameter is highlighted in the GUI_OUTPUT group.

Group	Alphabetical	DBD Order
GUI_COMMON		GUI_COMMON
DESC		Temperature Dem...
ASG		
UDF		1
GUI_LINKS		GUI_LINKS
DTYP		Soft Channel
FLNK		
GUI_INPUTS		GUI_INPUTS
SIOL		
SIML		
SIMS		<none>
GUI_OUTPUT		GUI_OUTPUT
VAL		
OUT		
OROC		
DOL		UserDemand

Comment

No object selected Frozen

Another graphical view of a Record



The small CapFast symbol for an Analogue
Output record



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





- ◆ *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 char's 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





- ◆ **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



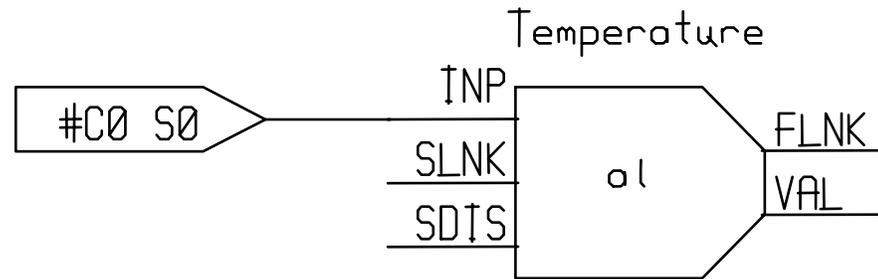
<i>INP</i>	<i>Input link</i>
<i>DTYP</i>	<i>Device type</i>
<i>RVAL</i>	<i>Raw data value</i>
<i>VAL</i>	<i>Engineering value</i>
<i>LOPR</i>	<i>Low operator range</i>
<i>HOPR</i>	<i>High operator range</i>

◆ *Analogue I/O records have these fields:*

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



Periodic Input



DTYP:XY566

SCAN:.1 Second

PHAS:0

LINR:LINEAR

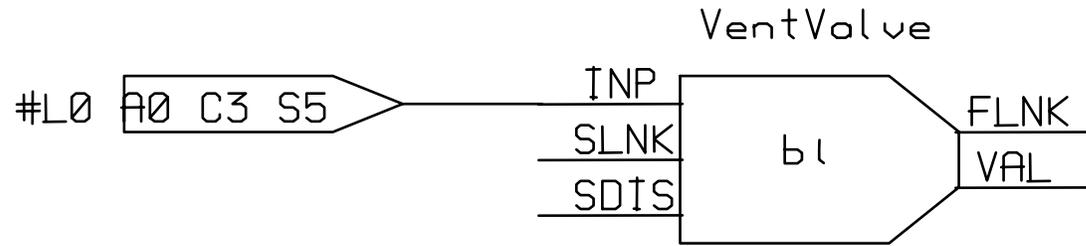
EGUL:0

EGUF:120

EGU:Celcius



Interrupt Input



DTYP:AB-Blnary Input

SCAN:I/O Intr

PHAS:0

ZNAM:Closed

ONAM:Open

ZSV:NO_ALARM

OSV:MAJOR_ALARM



Output records often have these fields



<i>OUT</i>	<i>Output link</i>
<i>DTYP</i>	<i>Device type</i>
<i>VAL</i>	<i>Engineering value</i>
<i>RVAL</i>	<i>Raw output value</i>
<i>DOL</i>	<i>Input link to fetch output value</i>
<i>OMSL</i>	<i>Output mode select:</i>

Supervisory, Closed Loop

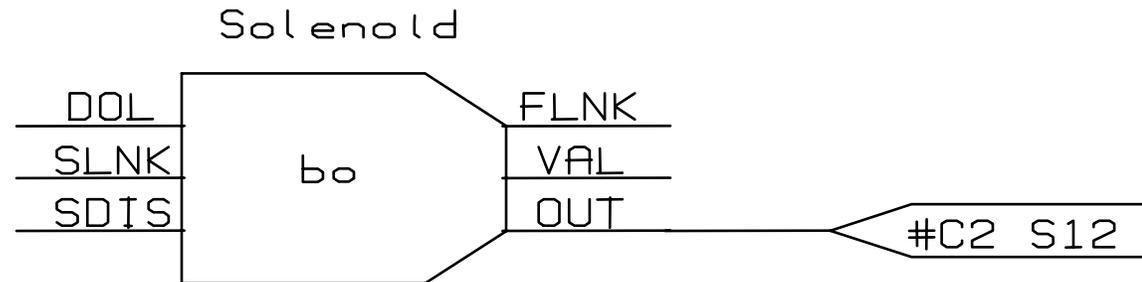
<i>LOPR</i>	<i>Low operator range</i>
<i>HOPR</i>	<i>High operator range</i>

◆ *Analogue outputs also have these fields:*

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



Passive Output



DTYP: XY220
SCAN: Passive
PHAS: 0
ZNAM: Locked
ONAM: Unlocked

- ◆ **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



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

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.



- ◆ **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

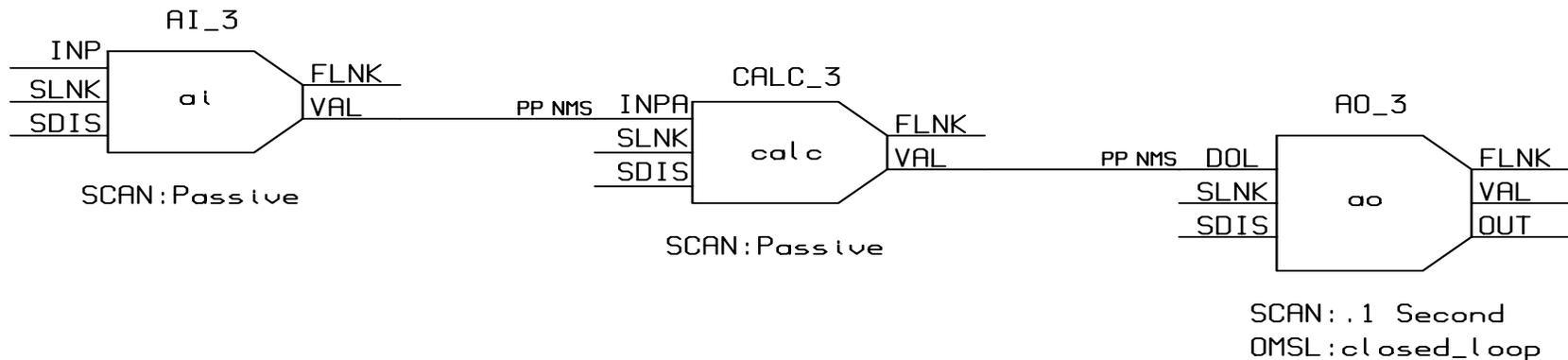
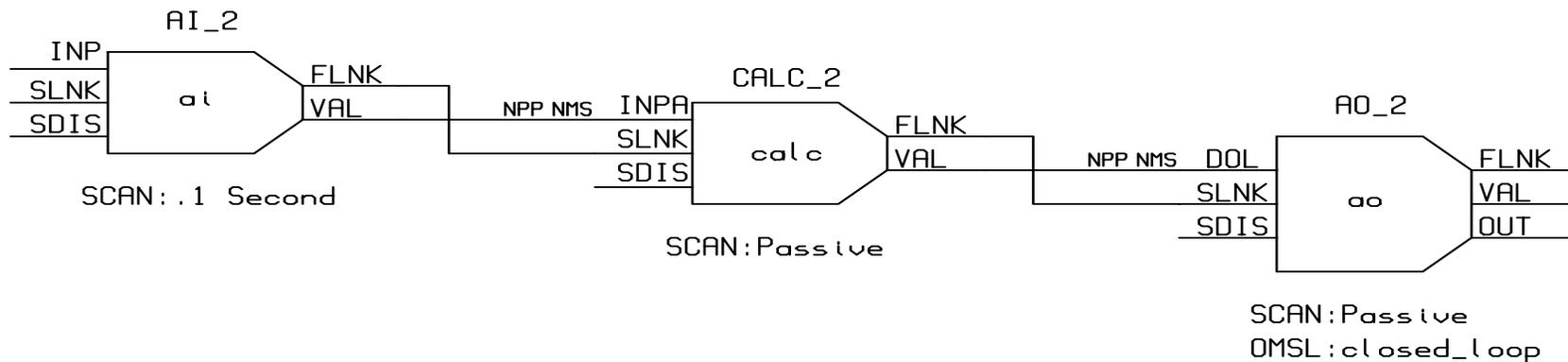
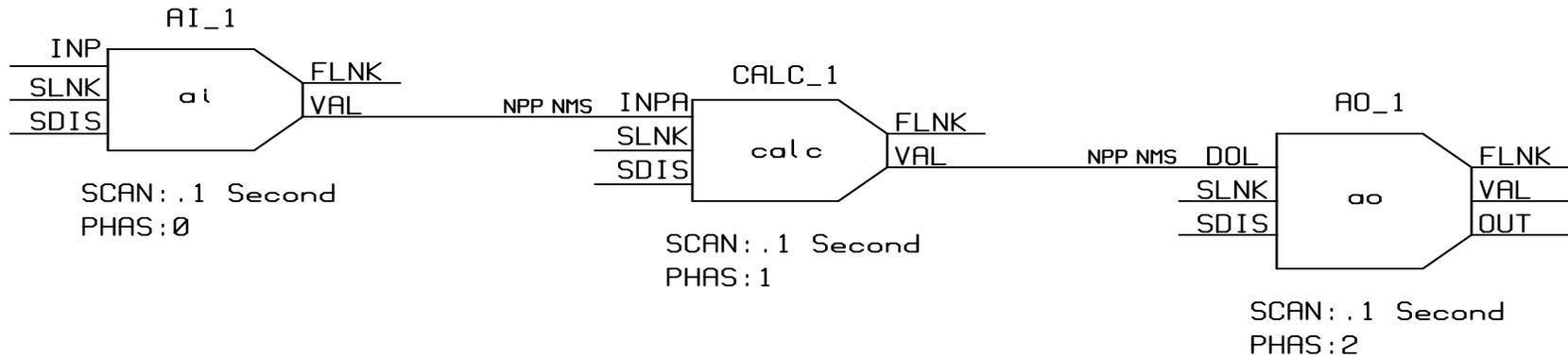
- ◆ “Hard” Input and Output records perform hardware I/O via device support
- ◆ “Soft” records access data from other records via DB or CA links
- ◆ 2 or 3 kinds of support are provided in recent R3.14 releases:
 - ❖ Soft Channel
 - ◆ *Get/Put VAL through link, no units conversion preformed*
 - ❖ Async 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



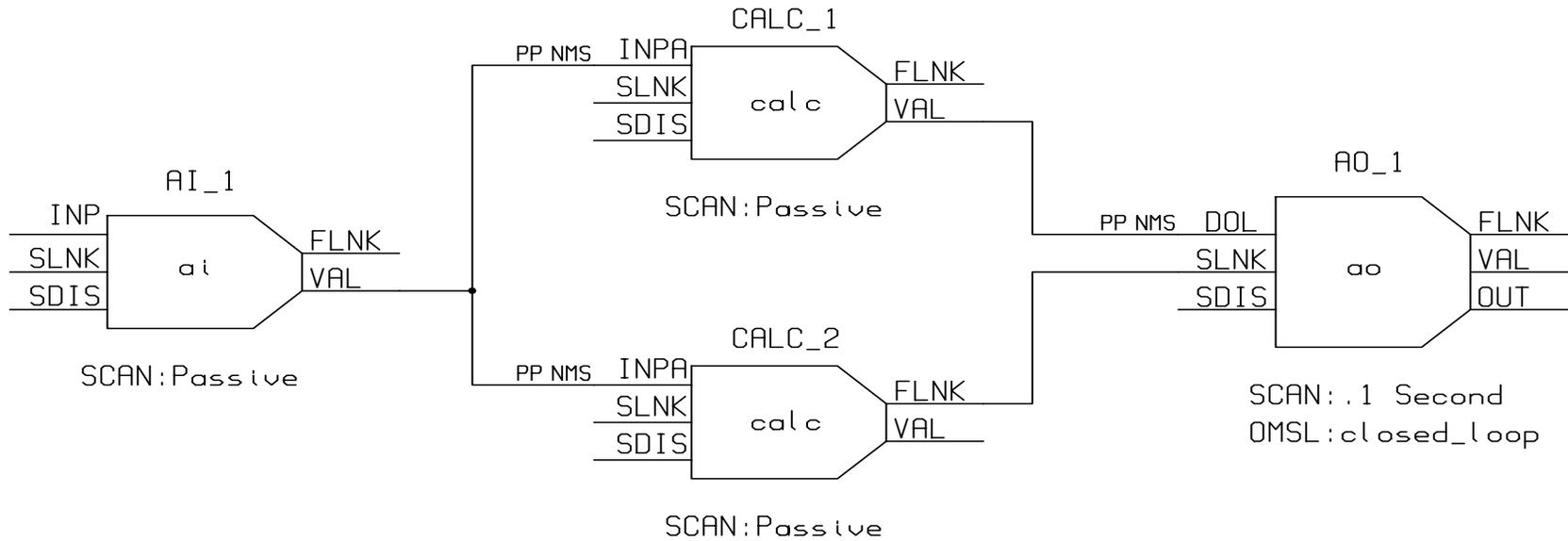
- ◆ Usually a Database link, referring to a record in same IOC
- ◆ Forward linking via Channel Access is possible, must explicitly name the PROC field of the remote record
- ◆ No flags (PP, NMS etc.)
- ◆ Destination record is only processed if it has
SCAN = Passive
- ◆ Does not pass a value, just causes subsequent processing



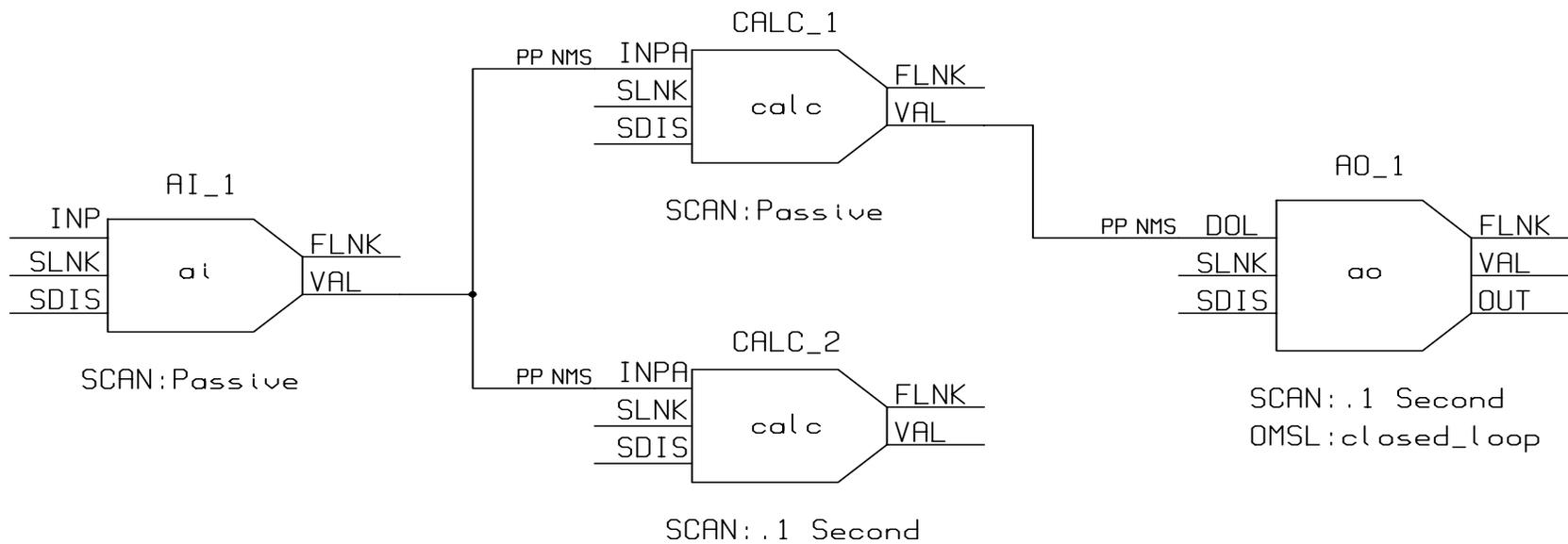
Processing chains



Which record is never processed?

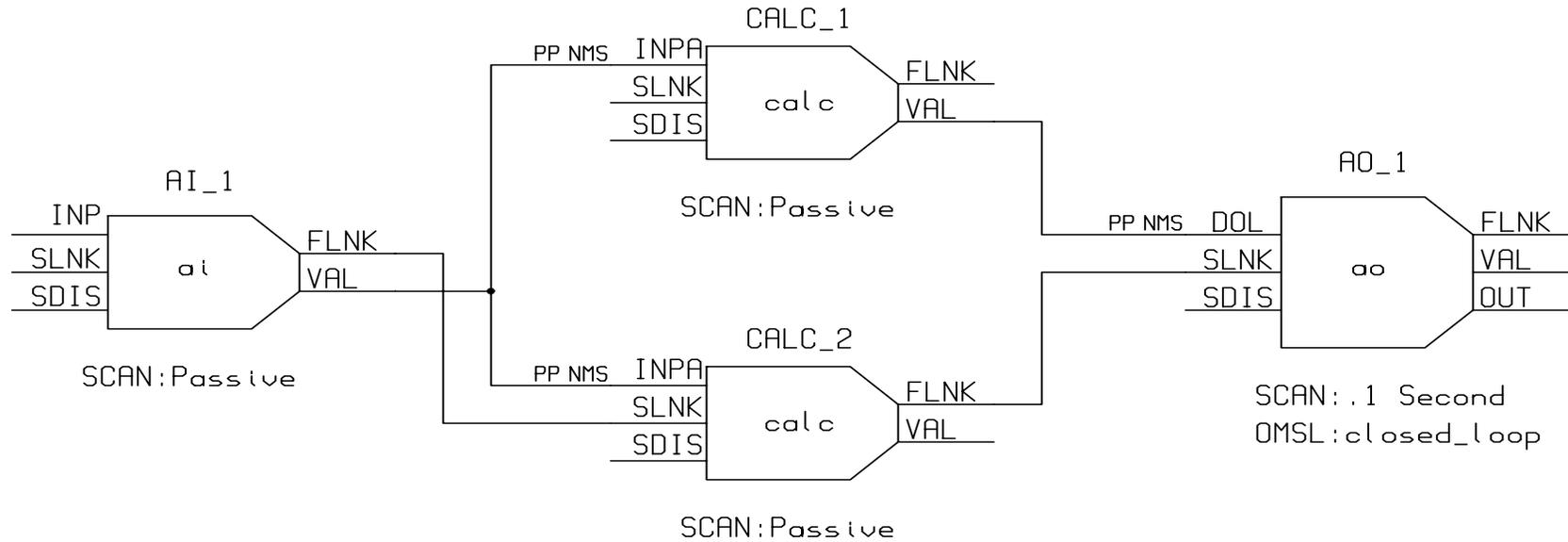


Which record is processed twice?



- ◆ **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

