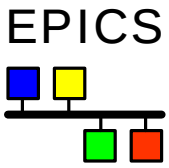
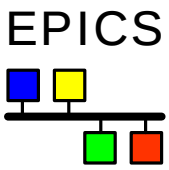


EPICS Database

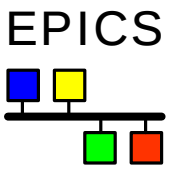
Marty Kraimer
Advanced Photon Source
Argonne National Laboratory

EPICS Database



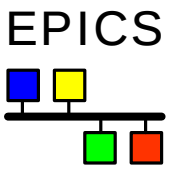
- References
  - Application Developer's Guide
  - Record Reference Manual
- Acknowledgement
  - This Presentation was adapted from Database Talks presented at the US Particle Accelerator School, July 1999, APS. The original presentations were prepared by Andrew Johnson and Ned Arnold

Database = Records + Fields + Links



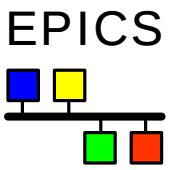
- An EPICS control system has IOCs
- Each IOC loads one or more Databases
- A Database is a collection of Records
- A Record is an object with:
  - A unique name
  - A behaviour based on it's record type
  - Controllable properties (fields)
  - Optional links to other records
- Record Support
  - Each record type has a Record Support Module
- Device Support
  - Each Record Type may have device support.
    - Multiple device support modules per record type
    - Each record instance has unique device support module

Record Activity



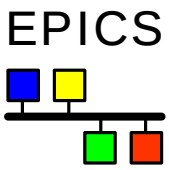
- Records are active — they can do things:
  - Get data from other records or from hardware
  - Perform calculations
  - Check values are in range & raise alarms
  - Put data to other records or to hardware
  - Activate or disable other records
  - Wait for hardware signals (interrupts)
- What a record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

Input Records



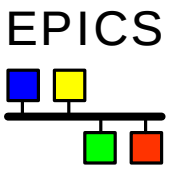
- Analog in
  - Read analog value, convert to engineering units, four alarm levels, simulation mode
- Binary in
  - Single bit, two states, assign strings to each state, alarm on either state or change of state, simulation mode
- Multi-bit binary in
  - Multiple bit, sixteen states, assign input value for each state, assign strings to each state, assign alarm level to each state, simulation mode
- String in
- Long integer
- Waveform
  - Configurable data type and array length
- …

Algorithms/Control Records



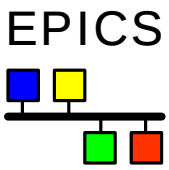
- Calc
  - 12 input links, user specified “calc expression”
  - Example: `(A-B)*C`
- Select
  - 12 input links, four select options
- Compress
  - Input link can be scalar or array
  - Algorithms include N to 1 compression (highest, lowest, or average), circular buffer of scalar input
- Subroutine
  - 12 input links, user provided subroutine, four alarm levels
- Fanout
  - Forward links to six other records
- …

Output Records



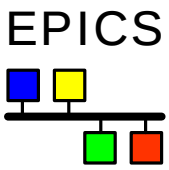
- Analog out
  - Write analog value, convert from engineering units, four alarm levels, closed_loop mode, drive limits, output rate-of-change limit, INVALID alarm action, simulation mode
- Binary out
  - Single bit, two states, assign strings to each state, alarm on either state or change of state, closed_loop mode, momentary 'HIGH', INVALID alarm action, simulation mode
- Multi-bit binary out
  - Multiple bit, sixteen states, assign output value for each state, assign strings to each state, assign alarm level to each state, closed_loop mode, INVALID alarm action simulation mode
- Long out
  - Write long integer value, four alarm levels, closed_loop mode, INVALID alarm action, simulation mode
- String out
  - Write a character string (40 max), closed_loop mode, INVALID alarm action, simulation mode
- …

Fields are for...



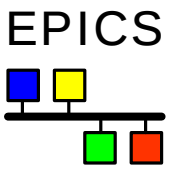
- Defining
  - What causes a record to process
  - Where to get / put data from / to
  - How to turn raw I / O data into a numeric engineering value
  - Limits indicating when to report an alarm
  - When to notify value changes to a client monitoring the record
  - Anything else which needs to be set 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



- Integers
  - char, short or long
  - signed or unsigned
- Floating-point numbers
  - float or double
- Strings
  - max length 40 characters or less
- Menu choices
  - select one from several 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



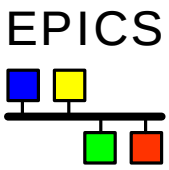
Design fields

NAME	28 Character unique name
DESC	28 Character description
ASG	Access security group
SCAN	Scan mechanism
PHAS	Scan order (phase)
PINI	Process at startup?
PRIO	Scheduling priority
SDIS	Scan disable input link
DISV	Scan disable value
DISS	Disabled severity
FLNK	Forward link

Run-time fields

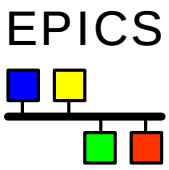
PROC	Force processing
PACT	Process active
STAT	Alarm status
SEVR	Alarm severity
TPRO	Trace processing
UDF	Set if record value undefined
TIME	Time when last processed

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

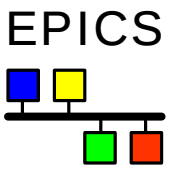
- Analog records have these fields:

LINR	Unit conversion control

  - No conversion, Linear, breakpoint tables…

EGUL	Low engineering value
EGUF	High engineering value
EGU	Engineering unit string

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

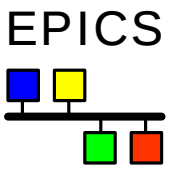
- Supervisory, Closed Loop

LOPR	Low operator range
HOPR	High operator range

- Analog outputs also 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

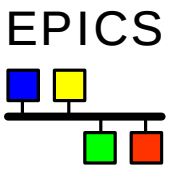
Links



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

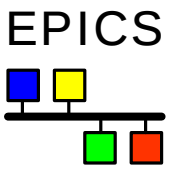
- Input link
  - Fetches data
- Output link
  - Writes data
- Forward link
  - Points to the record to be processed once this record finishes processing

Input and Output links may be...



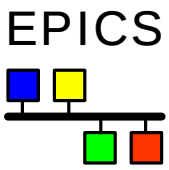
- Constant numeric value, eg:
  - 0
  - 3.1415926536
  - 1.6e-19
- Hardware link
  - A hardware I/O signal selector, the format of which depends on the device support layer
  - Several bus types are defined, e.g. VME_IO
- Process Variable link — the name of a record, which at run-time is resolved into
  - Database link
    - Named record is in this IOC
  - Channel Access link
    - Named record not found in this IOC

Device Support



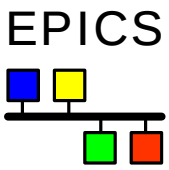
- Records do not access hardware directly
- The Device Support layer performs I/O operations on request
- A particular device support provides I/O for a single record type
- The DTYP field determines which device support to use
- The device support selected determines the format of the link (INP or OUT 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)

Soft Device Support



- Input and Output records are designed to perform hardware I/O via device support
- They can also access other records via DB or CA links, using soft device support
- 2 kinds of support are provided:
  - Soft Channel
    - Get/Put VAL through link, no conversion
  - Raw Soft Channel
    - Inputs
      - Get RVAL via input link
      - Convert RVAL to VAL (device specific)
    - Outputs
      - Convert VAL to RVAL (device specific)
      - Put RVAL to output link

Database Links



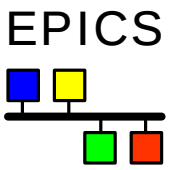
Components

- 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`

Forward links

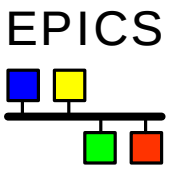


- Usually a database link refering to a record in same IOC
- Channel Access links possible, must name the `PROC` field of the remote record
- No flags (`.PP`, `.NMS` etc)
- Destination record must have

  `SCAN = Passive`

  for it to be processed
- Does not pass a value, just causes subsequent processing

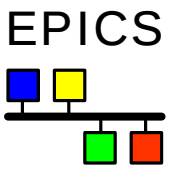
How are records given CPU time?



Several vxWorks tasks are used:

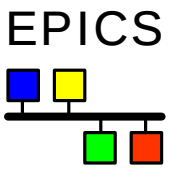
- callback (3 priorities) — I/O Interrupt
- scanEvent — Soft Event
- scanPeriod — Periodic
  - A separate task is used for each scan period
  - Faster scan rates are given higher vxWorks task priority
- Channel Access tasks use lower priority than record processing
  - If a CPU spends all the time doing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

Alarms



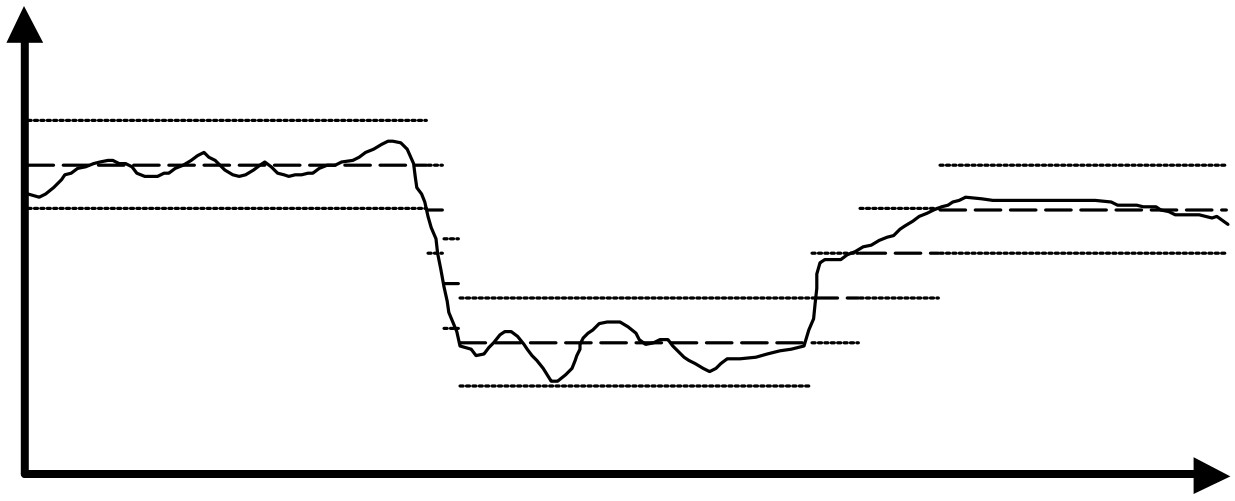
- Every record has the fields
  - SEVR Alarm Severity
    - NONE, MINOR, MAJOR, INVALID
  - STAT Alarm Status (reason)
    - READ, WRITE, UDF, HIGH, LOW, STATE, COS, CALC, DISABLE, etc.
- Most numeric records check VAL against HIHI, HIGH, LOW and LOLO fields after the value has been determined
- The 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

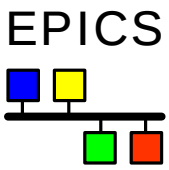


Channel Access notifies clients which are monitoring a numeric record when

- 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)



Simulation

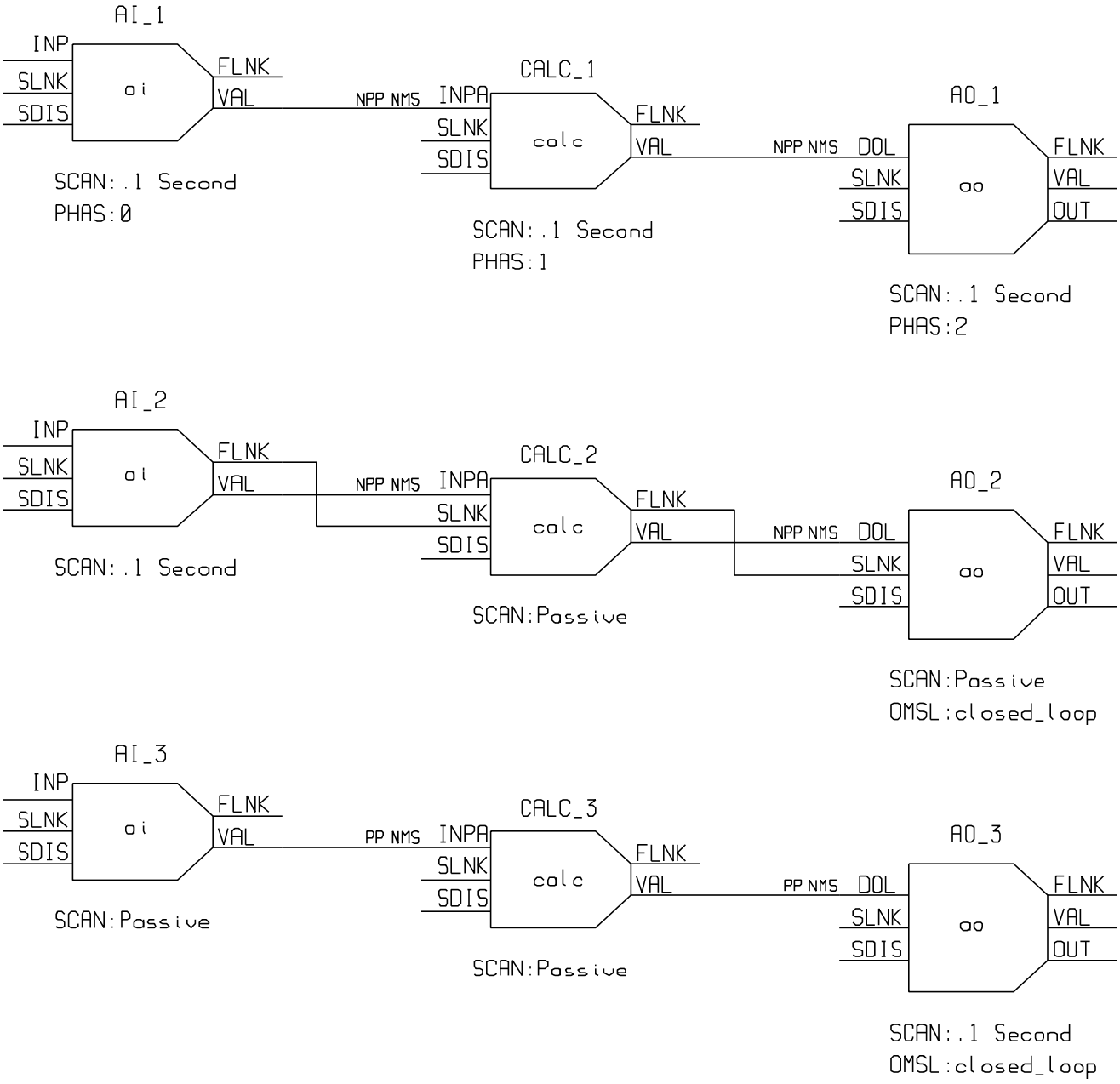


- Input and output record types often allow simulation of hardware interfaces

SIML	Simulation mode link
SIMM	Simulation mode value
SIOL	Simulation input link
SIMS	Simulation alarm severity

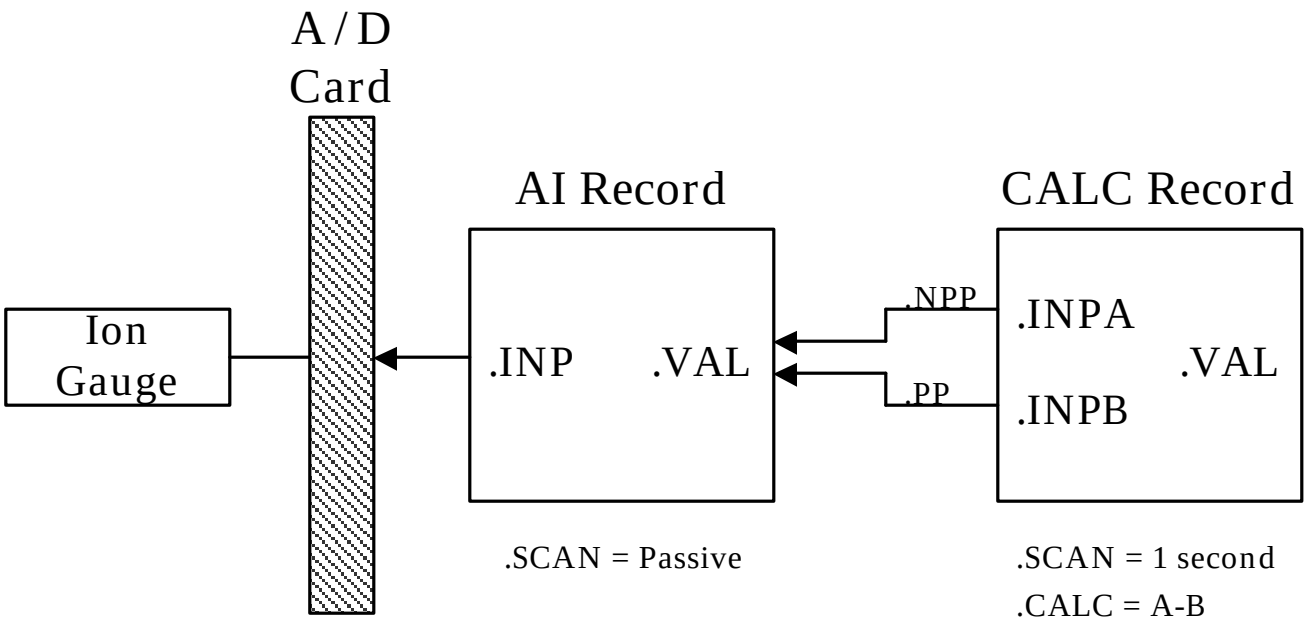
- Before using its device support, a record reads SIMM through the SIML link
- If SIMM=YES, device support is ignored; record I/O uses the SIOL link instead
- An alarm severity can be set whenever simulating, given by SIMS field

Example: Processing chains



Database Examples

Calculating “Rate-of-Change” of an Input

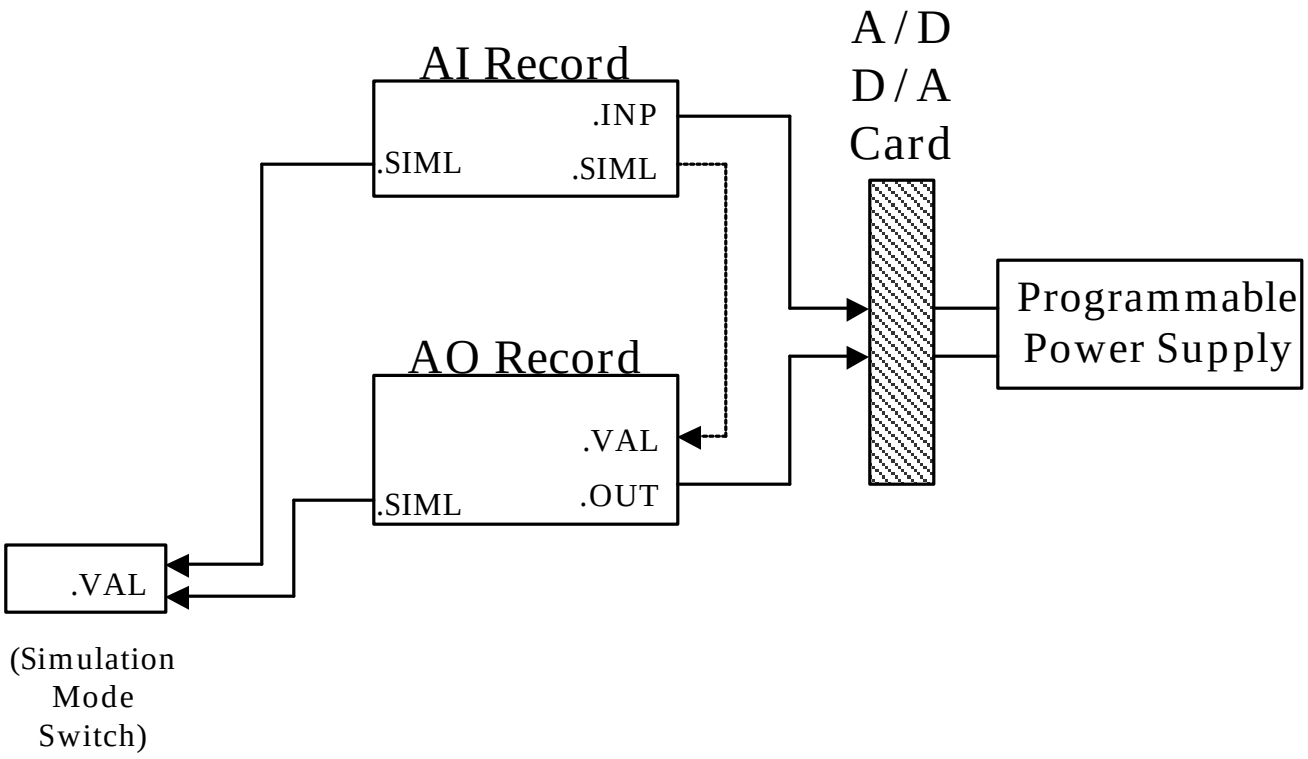


INPA fetches data that is 1 second old because it does not request processing of the AI record. INPB fetches current data because it requests the AI record to process. The subtraction of these two values reflects the ‘rate of change’ (difference / sec) of the pressure reading.

* The direction of the arrows indicates where a link points to, not necessarily the direction of the data flow.

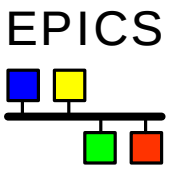
Database Examples

Simulation Mode



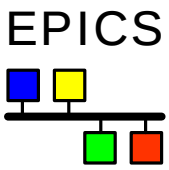
When in simulation mode, the AO record does not call device support and the AI record fetches its input from the AO record.

Defining the Database



- How does an IOC know what record types and device support options are available ?
  - Record types, device support options, enumerated menus, and other configuration options are defined in “database definition files” (.dbd)
  - During the IOC booting process, one or more .dbd files are loaded
  - .dbd files are created on the workstation to include the desired information for that IOC.
- How does an IOC know about record instances (the user’s database) ?
  - Record instances are describe in “database files” (.db)
  - During the IOC booting process, one or more .db files are loaded
  - .db files are created on the workstation to include the desired information for that IOC.

Example Db File

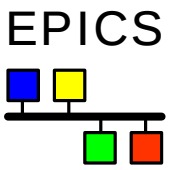


Analog Output Record

```
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 at run-time

Loading Database Files into the IOC

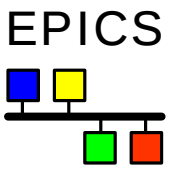


- Part of a typical startup script (st.cmd)

```
dbLoadDatabase("../../dbd/linacApp.dbd")
dbLoadRecords("../../db/xxLinacSim.db","user=studnt1")
iocInit
```

- One or more database definition files (.dbd) must be loaded first.
- Any record type specified in the database files must have been defined in the definition file
- Macros (variables) within the database files (e.g. $(user)) can be specified at boot time. This allows the same database to be loaded with different names or channel assignments.

Creating Database Files



- Since the database file is a simple ascii file, it can be generated by numerous applications … as long as the syntax is correct.
  - Text editor
  - Script
  - Relational Database Tool
  - EPICS-aware Database Configuration Tools
    - CAPFAST (a schematic entry application)
    - GDCT
    - JDCT
    - VDCT – May replace CAPFAST, GDCT, JDCT
- An EPICS-aware tool will read the .dbd file (library provided) and provide menu selections of enumerated fields. It may also detect database errors prior to the boot process
- A graphical tool is helpful to document and support complex databases