EPICS Tutorial

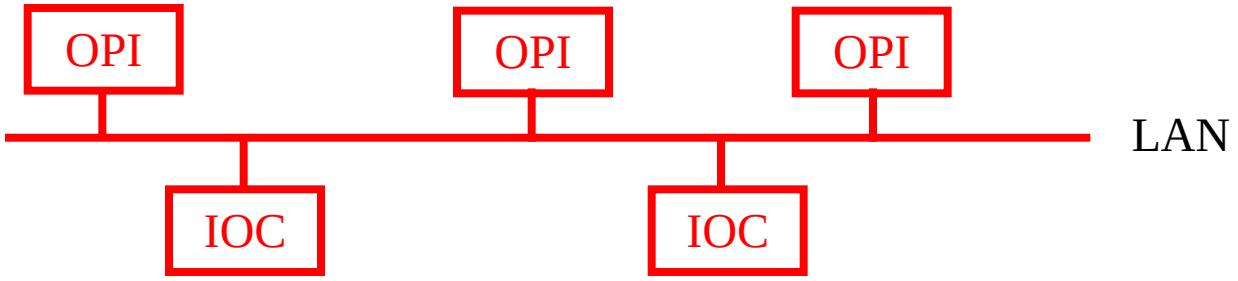
Experimental Physics and Industrial Control System

(EPICS)

Introduction to EPICS

The EPICS toolkit consists of a set of software components with which Application Developers can create a control system. The basic component types are:

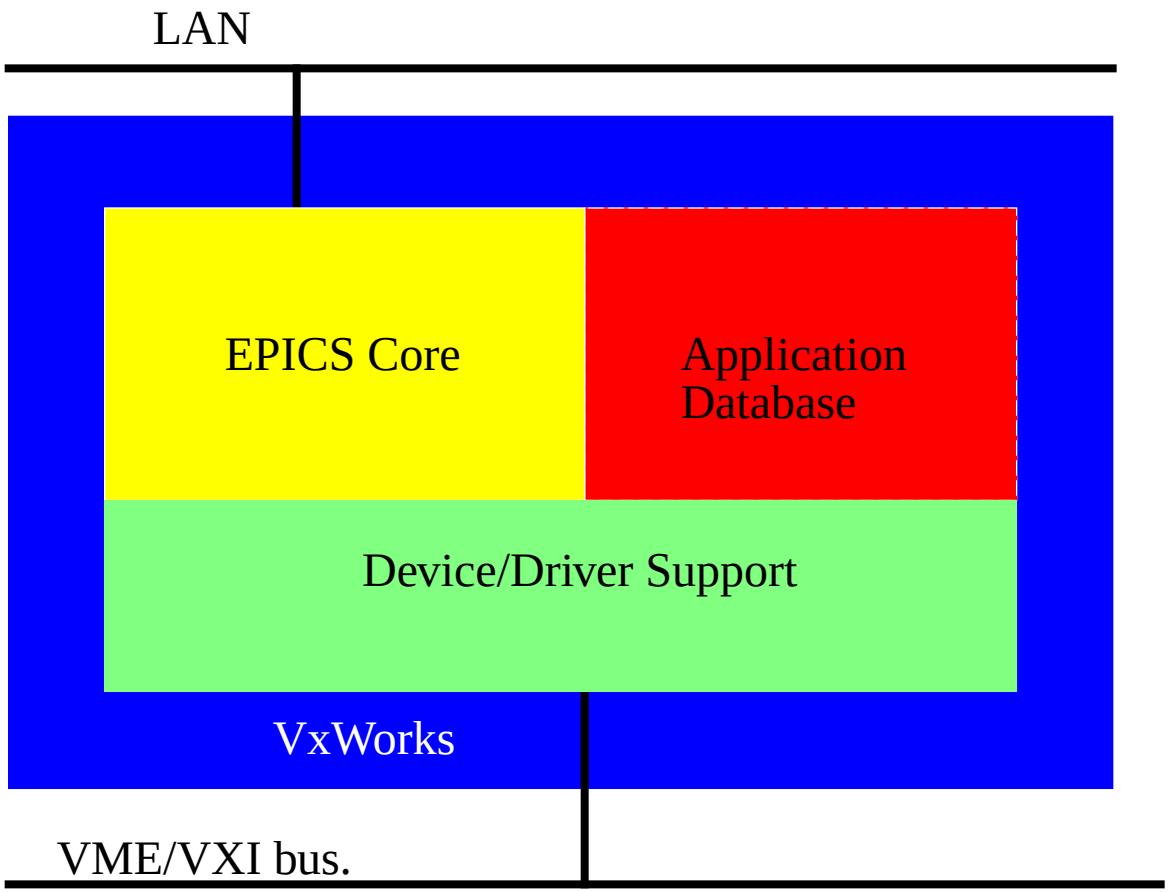
- OPI — **Operator Interface**. A UNIX- or NT-based workstation or PC which can run various EPICS tools—the "clients."
- IOC — I**nput Output Controller**. A VME/VXI-based chassis containing a Motorola 68K or PPC processor with various VME I/O modules for analog and digital signals, and for access to field buses such as Allen-Bradley, GPIB, CANbus or CAMAC.
- LAN — TCP/IP-based **Local Area Network**. A communication network which connects the IOCs and OPIs. EPICS provides a software component, Channel Access, which provides network transparent communication between every client—such as OPI—and an arbitrary number of servers—such as IOC.



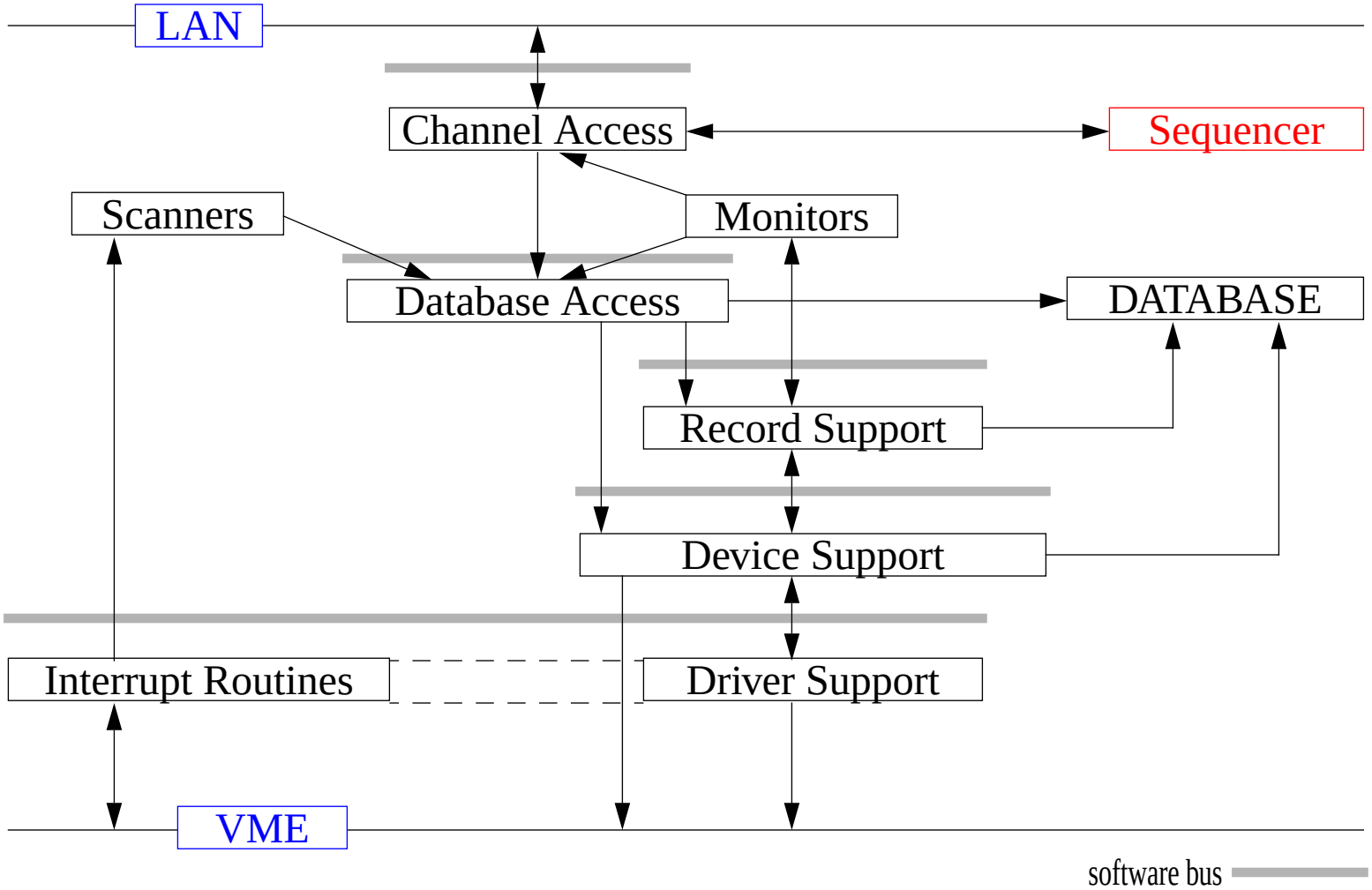
EPICS Attributes

• Tool Based	EPICS provides a number of tools both for creating and also for operating a control system. This minimizes the need for custom coding and helps ensure uniform operator interfaces.
• Distributed	An arbitrary number of IOCs and OPIs can be supported. If a single IOC becomes saturated, its functions may be spread over several IOCs.
• Event Driven	Network loading and message latency are simultaneously minimized by allowing only changes to a datum to be sent by servers to clients (“reporting by exception”).
• Customization	Users may add new components to support their applications; they are configured using simple ascii tables.
• Scalable	Systems from a few to ~100,000 “channels” supported.
• Modular	Distinct “software buses” at several layers allow good decoupling between clients and servers, across versions, and mixed hardware.
• Platforms	Runs on most Unix workstations, VMS, Windows NT.

IOC Components Overview



IOC Components.



IOC Database

The EPICS function-block database is the “heart” of the control system. It is:

- Application-specific.
- Memory-resident.
- What defines the control application; most other application are adjuncts to it.
- Modular: built by creating and linking many function blocks (“records”), selected from a user-expandable library.
- Deterministic: it runs in real-time either synchronously or asynchronously; either periodically or event-driven.
- Provided with fine-grained access control based on user, location, and dynamic parameters.
- Able to simulate missing hardware.

Record Support

- Has templates of the processing to be performed by each record instance.
- Processing dependent upon record type.
- For I/O records, Record support calls Device support.
- Record support does not know any details of Device support, only how to invoke it.

Generally, record support will perform some combination of the following:

• I/O	If record is an I/O type, Read or Write to hardware via DevSup.
• Conversion	Conversion of raw data to user defined units (with smoothing, scaling, masking, shifting).
• Alarms	Check for and raise alarms (high, low).
• Monitor	Trigger monitors (callbacks).
• Link	Cause processing of related records, possibly by “pulling” or “pushing” data.

Device and Driver Support

Device and Driver Support used by I/O type records.

Device Support:

- Processing is dependent upon device type (Allen-Bradley, VME ADC, *etc*.).
- Calling record's attributes are obtained though a passed pointer.
- If hardware access is complicated, it calls Driver Support for low-level I/O.
- "Soft" types (available for most record types) for simulation, mailboxes, *etc*.

Driver Support:

- Provides data hiding from DevSup, handles low level bus I/O.
- Its inclusion is at the application designer's discretion.

Database Access

- All other EPICS components access a record eventually via dbAccess.
- External units reach dbAccess via Channel Access, including records on other IOCs.
- Internal units (other records on same IOC) access a record directly through dbAccess, but can optionally use Channel Access.
- dbAccess allows read/write operations on a record's attributes.

Scanners

The Scanners define when a record will be processed, and by whom.

Records are processed in one of three ways:

- Periodic — At a certain periodicity, *e.g.*, 1 second, 0.1 second, *etc*.
- Event — Processed when a software or hardware event is triggered.
- Passive — Executed because a related record was invoked (via a link), or a field was updated externally (via a dbPut).

Within VxWorks, records and their support code (recSup, devSup and drvSup) are processed by a scanner task (a *thread*). There are several tasks (configurable) which can perform the processing; the one which is used depends upon the record's scan attribute. Types are:

- Periodic tasks of varying priority.
- General purpose tasks for event driven records and Interrupt Service Routine callbacks; high, medium and low priorities.

Channel Access

Channel Access is the “backbone” of EPICS. It connects internal IOC components (database) and external components. External components may actually reside on the same IOC, such as a Sequencer, but normally will reside on a remote IOC or OPI host.

Channel Access is based on a Client/Server model.

Each IOC runs its own local Channel Access Server tasks.

Operations allowed by a remote Channel Access Client include:

- Search the network for a particular Channel* and connect to it.
- Write to that Channel.
- Read from that Channel.
- Post a monitor on (await callback from) that Channel.
- Disconnect from that Channel.

Channel Access runs over the network using TCP/IP for data transport and UDP/IP for connection management. In addition to the “value” of a channel, a timestamp, a status, and a severity are returned.

* A Channel is defined as <record name>.<field name>.

Sequencer

Provides Finite-State Machine paradigm for control of the database(s). Source code is written in State Notation Language (SNL) which is compiled into C code using the EPICS State Notation Compiler (SNC). *Runs directly on IOCs*.

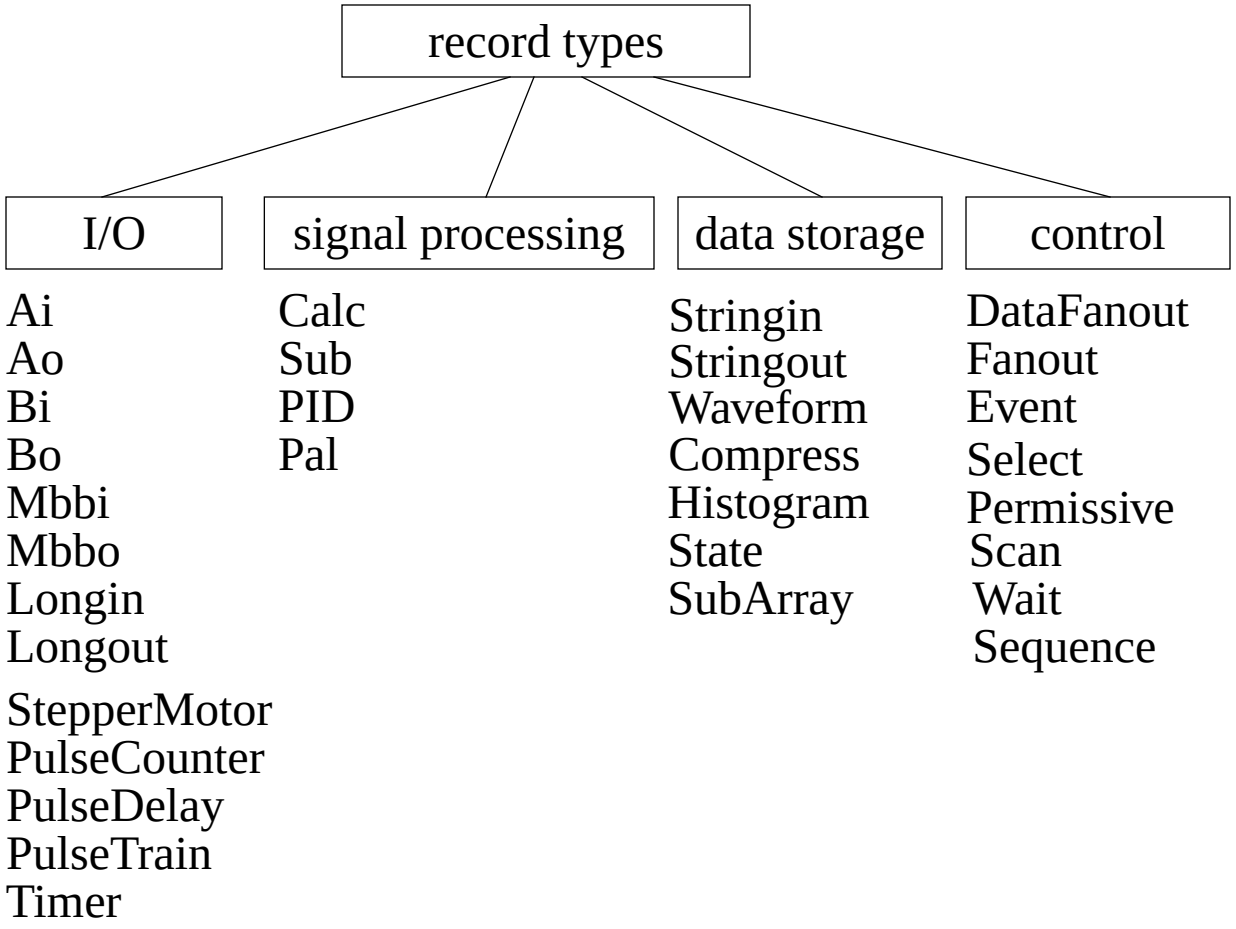
State transitions are triggered by: elapsed time, channel value change, software event.

SNL allows easy translation from State Diagrams to source code. Example:

```
State A {
          when (situation X occurs){
                handle situation X;
          } State B;
          when (situation Y occurs){
                handle situation Y;
          } State C;
}
...
```

Record Types

Record types fall into four basic categories:



Record Fields

A record's attributes are defined by its fields. A field name is abbreviated to four or less letters and is conventionally written in capitals. As fully qualified it is written:

<record name>.<field name>

and is referred to as a *Channel*.

All record types inherit a core set of fields which are needed for basic record processing, such as:

- PROC Writing to this field causes record processing.
- SCAN The record's method of scanning.
- VAL The record's value*.
- FLNK The record's processing linkage to other records.

Specific record types inherit the fields for that type, *e.g.*, an Analog Input record has:

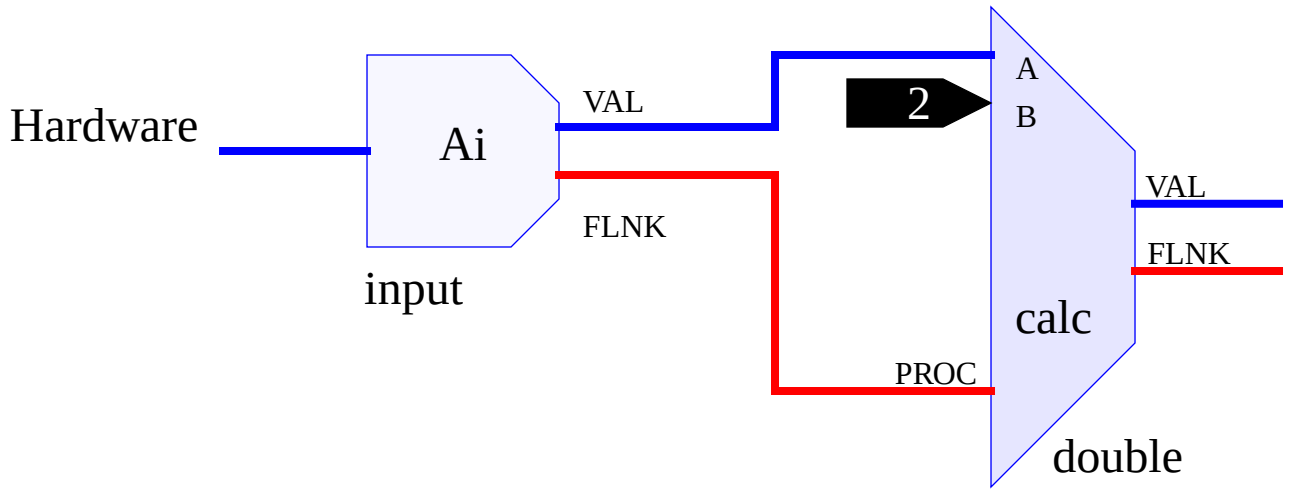
- DTYP The I/O Device Type.
- INP The hardware address.

Record processing code (RecSup, DevSup, DrvSup) accesses the fields of a record instance via a pointer.

* If no field is given in a Channel, the VAL field is defaulted.

Record Linking

Records in a database may by linked for purposes of data and processing flow.



Data Flow

Process Flow

double.CALC = A*B

OPI Tools

OPI Tools run on UNIX Workstations (Sun, HP, SGI, Alpha), Windows NT PCs, and some others.

They are executed at run time, providing dynamic control/monitoring of the system. No traditional programming is required to configure them.

- EDD/DM — Display Manager: GUI for control/monitoring.
- MEDM — Another Display Manager.
- ALH — Alarm Handler; graphical display of alarm tree.
- AR/ARR — Archiver; recording/playback of sensor data.
- BURT — Backup/Restore Tool; snapshots of setpoints.
- KM — Knob Manager; attaches physical dials to setpoints.
- DP — Display Page; tabular arrangement of channels.
- StripTool — Dynamic strip chart display.

Configuration Tools

- GDCT — Graphical Database Configuration Tool to generate database structures.
- Capfast — Schematic editor to generate hierarchical database structures.
- DCT — Database Configuration Tool.

Almost all EPICS configuration files (such as the Database) have straight-forward ASCII formats and are easily generated by report generators, scritps, text editors, and other host tools.

General Access

A number of general-purpose commercial and public-domain facilities have been enabled with a Channel Access capability, including:

- Unix shells
- Tcl/Tk — Interpreter with GUI
- Perl — Scripting Language
- Mathematica — Modelling
- Wingz — Spreadsheet
- IDL/PV-Wave — Presentation Graphics
- Labview — Virtual Instrument Builder
- Java — Object-oriented application builder (via Gateway)

Utilities

On OPI:

- Probe — Single Channel GUI Diagnostic
- XMCA — Multi-Channel GUI Diagnostic
- CAU — Channel Access Test Utility
- ioc_stats — IOC statistics utility

In IOC:

- db... — Database Utilities (get, put, dump, trace, *etc*.)
- dbior — Driver Reports
- as.. — Access Security Reports
- seq... — Sequencer Reports
- csar — Channel Access Reports