More Record Details

BaBar-ESD Training

May, 2002
One view of a Record

The small CapFast symbol for an Analogue Output record
Another view

<table>
<thead>
<tr>
<th>DESC</th>
<th>STRING</th>
<th>Descriptor</th>
<th>Sea Temperature</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASG</td>
<td>STRING</td>
<td>Access Security Group</td>
<td></td>
</tr>
<tr>
<td>SCAN</td>
<td>MENU</td>
<td>Scan Mechanism</td>
<td>1 second</td>
</tr>
<tr>
<td>PINI</td>
<td>MENU</td>
<td>Process at icInit</td>
<td>NO</td>
</tr>
<tr>
<td>PHAS</td>
<td>INTEGER</td>
<td>Scan Phase</td>
<td>0</td>
</tr>
<tr>
<td>EVNT</td>
<td>INTEGER</td>
<td>Event Number</td>
<td>0</td>
</tr>
<tr>
<td>TSEL</td>
<td>INLINK</td>
<td>Time Stamp Link</td>
<td></td>
</tr>
<tr>
<td>DTPV</td>
<td>DEVICE</td>
<td>Device Type</td>
<td>vsWorks Variable Form</td>
</tr>
<tr>
<td>OUT</td>
<td>OUTLINK</td>
<td>Output specification</td>
<td>@sense1 Form</td>
</tr>
<tr>
<td>DISV</td>
<td>INTEGER</td>
<td>Disable Value</td>
<td>1</td>
</tr>
<tr>
<td>SDIS</td>
<td>INLINK</td>
<td>Scanning Disable</td>
<td></td>
</tr>
<tr>
<td>DISS</td>
<td>MENU</td>
<td>Disable Alarm Severity</td>
<td>NO_ALARM Form</td>
</tr>
<tr>
<td>PRIO</td>
<td>MENU</td>
<td>Scheduling Priority</td>
<td>LOW Form</td>
</tr>
<tr>
<td>FLNK</td>
<td>FWDLINK</td>
<td>Forward Process Link</td>
<td>0 Form</td>
</tr>
<tr>
<td>OROC</td>
<td>REAL</td>
<td>Output Rate of Change</td>
<td>0.1 Form</td>
</tr>
<tr>
<td>DOL</td>
<td>INLINK</td>
<td>Desired Output Loc</td>
<td></td>
</tr>
<tr>
<td>OMSL</td>
<td>MENU</td>
<td>Output Mode Select</td>
<td>supervisory Form</td>
</tr>
<tr>
<td>OIF</td>
<td>MENU</td>
<td>Out Full/Incremental</td>
<td>Full Form</td>
</tr>
<tr>
<td>PREC</td>
<td>INTEGER</td>
<td>Display Precision</td>
<td>1</td>
</tr>
<tr>
<td>LNR</td>
<td>MENU</td>
<td>Linearization</td>
<td>NO CONVERSION Form</td>
</tr>
<tr>
<td>EQUF</td>
<td>REAL</td>
<td>Eng Units Full</td>
<td>0</td>
</tr>
<tr>
<td>EQUL</td>
<td>REAL</td>
<td>Eng Units Low</td>
<td>0</td>
</tr>
<tr>
<td>EQU</td>
<td>STRING</td>
<td>Engineering Units</td>
<td>calc1us</td>
</tr>
<tr>
<td>DRVH</td>
<td>REAL</td>
<td>Drive High Limit</td>
<td></td>
</tr>
<tr>
<td>DRVL</td>
<td>REAL</td>
<td>Drive Low Limit</td>
<td></td>
</tr>
<tr>
<td>HOPR</td>
<td>REAL</td>
<td>High operating Range</td>
<td></td>
</tr>
<tr>
<td>LOPR</td>
<td>REAL</td>
<td>Low operating Range</td>
<td></td>
</tr>
<tr>
<td>AOFF</td>
<td>REAL</td>
<td>Adjustment Offset</td>
<td>0</td>
</tr>
</tbody>
</table>

Close
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(PRIQ,"LOW")
    field(FLNK,"")
    field(OUT,"#C0 S0")
    field(CROC,"0.0e+00")
    field(DOL,"")
    field(OSM,"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(SIO,"")
    field(SML,"")
    field(SMS,"NO_ALARM")
    field(IMC,"Continue normally")
    field(IVQ,"0.0e+00")
}
```

This shows only the design fields, there are other fields which are used at run-time.
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
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
Which record is never processed?
Which record is processed twice?
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?
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
Lock-sets

- Prevent a record from being processed simultaneously from two scan tasks
- A lock-set is a group of records interconnected by:
  - Output Database links
  - Forward links
  - Input links which are .PP or .MS
  - Arrays
- Lock-sets are determined automatically by the LOC at start-up

You can split a lock set with

- Channel Access links, using .CA flag
- Database links which are .NPP .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.
 le}.

- 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).
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**)
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.
Application Structure

```
ronc@flora04 $ type st.cmd
dbLoadDatabase(".././db/example.db",0,0)
registerRecordDeviceDriver(pdbname)
dbLoadRecords(".././db/dbExample.db","user=roncHost")
dbLoadRecords(".././db/dbExample2.db","user=roncHost,no=1,scan=1 second")
dbLoadRecords(".././db/dbExample2.db","user=roncHost,no=2,scan=2 second")
dbLoadRecords(".././db/dbExample2.db","user=roncHost,no=3,scan=5 second")
iocInit()
#seq sncExample_"user=roncHost"
```
Steps to Creating and Loading a New Database File

- Create the database file in an appropriate Db directory
- Edit the Makefile.Host so this .db file is managed properly
- gnumake
- Edit the IOC’s startup script (st.cmd) to load the new database
  - dbLoadRecords(______)
- Reboot the IOC
VDCT

- Start *vdct* (usually in a Db directory)
- Open one or more .dbd files (usually in the directory ‘../../dbd’) to define available record types, menus, available device options, etc)
- Create, copy, edit record instances
- Save the .db file