

EPICS: javaIOC

Marty Kraimer
EPICS Collaboration Meeting
ANL/APS June 12–16 2006

Summary

- Background
- Overview of JavalOC
- Current Status
- Remaining work for a beta release
- Other desirable components
- Example XML definitions
- PV Naming

Background

- Ideas developed during 2005
 - Described in EPICS core-developer wikis
http://www.aps.anl.gov/epics/wiki/index.php/Core_Developer_Pages
 - V4 Database Definition
 - V4 Database Runtime
 - Many basic concepts are the same
 - Details quite different
 - Java only
 - XML for Database Definition & Instance
 - ...

javaIOC Overview

- **Syntax is XML:** Database Definitions and Record Instances
- **Field Types:** structure and array are supported
- **Channel access:** Similar for Clients and Database Links
- **Database**
 - Each field is object accessed via Interface
 - Record/link support access fields via interfaces
 - Database handles monitors without support calling `db_post_event`
- **Better support for data acquisition**

Current Status

- Database Implemented
 - Standard support for reflection
 - Default data implementation
 - support code can provide implementation
 - Listen support – called when field changes
- XML parsers implemented
 - Database Definitions & Record Instance
- Lots remaining before beta release

Current Status - Continued

- **Basic field types – used by CA**
 - Primitive + String: all Java primitives except char
 - boolean, byte, short, int, long, float, double, string
 - enum
 - array – element can be any basic type
 - structure – fields can be any basic type
- **Database extends types**
 - menu – enum with immutable choices
 - link – supportName + support structure
 - structure – fields can also be extended type
 - array – element type can be extended type

Beta Release: Remaining Tasks

- Channel access
 - Local and remote
 - Gateway to/from EPICS V3
- Record processing – see wiki for vision
- Include and macro substitution for XML
- Shell – probably Jython
- iocInit – will support on-line add

Beta Release: Remaining Tasks-continued

- Design and implement core set of
 - Database Definition XML files:
menu structure linkSupport recordType
 - record/structure support modules
 - channel access link support
 - hardware link support based on asynDriver
- Lots to do!

Other Desirable Components

- VDCT: Visual Database Configuration Tool
- Access security
- Error logging
- Put logging
- CA clients: Display Manager, Alarm Handler, etc.
- ???
- Lots and lots to do!!

Example XML Definitions

- Database Definitions
 - menu
 - structure
 - linkSupport
 - recordType
- Record Instance

menu definition

```
<menu name = "menuAlarmSevr">  
  <choice>none</choice>  
  <choice>minor</choice>  
  <choice>major</choice>  
  <choice>invalid</choice>  
</menu>
```

structure definition

```
<structure name = "displayLimit">  
  <field name = "low" type = "double" />  
  <field name = "high" type = "double" />  
</structure>
```

```
<structure name = "linearConvert"  
  structureSupportName = "org.epics.ioc.support.LinearConvert">  
  <field name = "engUnitsLow" type = "double" />  
  <field name = "engUnitsHigh" type = "double" />  
  <field name = "slope" type = "double" />  
  <field name = "intercept" type = "double" />  
</structure>
```

recordType definition

```
<recordType name = "example"  
  recordSupportName = "org.epics.ioc.support.ExampleRecordSupport"/>  
  <field name = "value" type = "double">  
  <field name = "priority" type = "menu" menuName = "menuPriority" />  
  <field name = "units" type = "string" />  
  <field name = "status" type = "string" />  
  <field name = "severity" type = "menu" menuName = "menuAlarmSevr" />  
  <field name = "timeStamp" type = "structure" structureName = "timeStamp" />  
  <field name = "displayLimit" type = "structure" structureName = "displayLimit" />  
</recordType>
```

record instance definition

```
<record name = "exampleInstance" type = "example"  
  recordSupportName = "MyExampleRecordSupport" >  
  <value>10.0</value>  
  <units>volts</units>  
  <displayLimit>  
    <low>0.0</low>  
    <high>10.0</high>  
  </displayLimit>  
</record>
```

recordType Examples

- Simple example was shown
- Build a type via embedded structures
 - Analog input example
 - Power supply example
- Embedding vs linked records
 - javaIOC supports both approaches
 - Design decision

Example: Analog Input

- recordType aiLinearRecord
 - Has a field that is a structure aiLinear
 - Support works with other structures
- structure aiLinear
 - Has a field that is a structure aiRaw
 - Support works with other structures
- structure aiRaw
 - Support gets input via asyn style interfaces

Structure aiRaw

```
<structure name = "aiRaw" structureSupportName = "org.epics.ioc.support.AiRaw" >  
  <field name = "value" type = "int" >  
    <property name = "status" associatedField = "status" />  
    <property name = "severity" associatedField = "severity" />  
    <property name = "timeStamp" associatedField = "/timeStamp" />  
  </field>  
  <field name = "status" type = "string" />  
  <field name = "severity" type = "menu" menuName = "menuAlarmSevr" />  
  <field name = "input" type = "link" />  
</structure>
```

Structure aiLinear

```
<structure name = "aiLinear"
  structureSupportName = "org.epics.ioc.support.AiLinear" >
  <property name = "rawValue" associatedField = "aiRaw.value" />
  <field name = "value" type = "double" >
    <property name = "status" associatedField = "status" />
    <property name = "severity" associatedField = "severity" />
    <property name = "units" associatedField = "units" />
    <property name = "displayLimit" associatedField = "displayLimit" />
    <property name = "timeStamp" associatedField = ".." />
  </field>
  <field name = "status" type = "string" />
  <field name = "severity" type = "menu" menuName = "menuAlarmSevr" />
  <field name = "units" type = "string" />
  <field name = "displayLimit"
    type = "structure" structureName = "displayLimit" />
  <field name = "aiRaw" type = "structure" structureName = "aiRaw"/>
  <field name = "linearConvert"
    type = "structure" structureName = "linearConvert" />
</structure>
```

recordType aiLinearRecord

```
<recordType name = "aiLinearRecord"  
  recordSupportName = "org.epics.ioc.support.AiLinear">  
  <property name = "value" associatedField = "aiLinear.value" />  
  <property name = "rawValue" associatedField = "aiLinear.rawValue" />  
  <property name = "status" associatedField = "aiLinear.status" />  
  <property name = "severity" associatedField = "aiLinear.severity" />  
  <field name = "aiLinear" type = "structure" structureName = "aiLinear" />  
  <field name = "priority" type = "menu" menuName = "menuPriority" />  
  <field name = "timeStamp" type = "structure" structureName = "timeStamp"/>  
  <field name = "scan" type = "structure" structureName = "scan"/>  
</recordType>
```

Record Instance aiLinearRecord

```
<record name = "exampleAiLinear" type = "aiLinearRecord">
  <aiLinear>
    <aiRaw>
      <input linkSupportName = "asynInt" configStructureName = "asynLink">
        <portName>somePort</portName>
        <addr>1</addr>
      </input>
    </aiRaw>
    <units>volts</units>
    <displayLimit>
      <low>0.0</low>
      <high>10.0</high>
    </displayLimit>
    <linearConvert>
      <engUnitsLow>0.0</engUnitsLow>
      <engUnitsHigh>10.0</engUnitsHigh>
    </linearConvert>
  </aiLinear>
  <priority>medium</priority>
</record>
```

Example: Power Supply

- structure `powerSupply`
 - Simple just power, current, voltage
 - current and voltage via structure `aiLinear`
 - power is just $\text{current} \times \text{voltage}$
- recordType `powerSupplyRecord`
 - Has a field which is structure `powerSupply`
- recordType `powerSupplyArrayRecord`
 - Has an array field with element type structure
 - Instance makes elements `powerSupply`

structure powerSupply

```
<structure name = "powerSupply"
  structureSupportName = "somepackage.support.PowerSupply">
  <property name = "current" associatedField = "currentInput.value" />
  <property name = "voltage" associatedField = "voltageInput.value" />
  <field name = "power" type = "double" >
    <property name = "status" associatedField = "status" />
    <property name = "severity" associatedField = "severity" />
    <property name = "units" associatedField = "units" />
    <property name = "displayLimit" associatedField = "displayLimit" />
    <property name = "timeStamp" associatedField = ".." />
  </field>
  <field name = "currentInput" type = "structure" structureName = "aiLinear" />
  <field name = "voltageInput" type = "structure" structureName = "aiLinear" />
  <field name = "status" type = "string" />
  <field name = "severity" type = "menu" menuName = "menuAlarmSevr" />
  <field name = "units" type = "string" />
  <field name = "displayLimit"
    type = "structure" structureName = "displayLimit" />
</structure>
```

recordType powerSupplyRecord

```
<recordType name = "powerSupplyRecord"  
  structureSupportName = "somepackage.support.PowerSupply">  
  <property name = "power" associatedField = "powerSupply.power" />  
  <property name = "current" associatedField = "powerSupply.current" />  
  <property name = "voltage" associatedField = "powerSupply.voltage" />  
  <field name = "powerSupply"  
    type = "structure" structureName = "powerSupply" />  
  <field name = "status" type = "string" />  
  <field name = "severity" type = "menu" menuName = "menuAlarmSevr" />  
  <field name = "timeStamp" type = "structure" structureName = "timeStamp"/>  
  <field name = "priority" type = "menu" menuName = "menuPriority" />  
  <field name = "scan" type = "structure" structureName = "scan"/>  
</recordType>
```

recordType powerSupplyArrayRecord

```
<recordType name = "powerSupplyArrayRecord"  
  structureSupportName = "somepackage.support.PowerSupplyArray" >  
  <field name = "powerSupply" type = "array" elementType = "structure" />  
  <field name = "status" type = "string" />  
  <field name = "severity" type = "menu" menuName = "menuAlarmSevr" />  
  <field name = "timeStamp" type = "structure" structureName = "timeStamp"/>  
  <field name = "priority" type = "menu" menuName = "menuPriority" />  
  <field name = "scan" type = "structure" structureName = "scan"/>  
</recordType>
```

PowerSupplyRecord instance

```
<record name = "examplePowerSupply" type = "powerSupplyRecord">
  <powerSupply>
    <units>watts</units>
    <displayLimit>
      <low>0.0</low>
      <high>100.0</high>
    </displayLimit>
    <currentInput>
      <aiRaw>
        <input linkSupportName = "asynInt" configStructureName = "asynLink">
          <portName>somePort</portName> <addr>2</addr>
        </input>
      </aiRaw>
      <units>amps</units>
      <displayLimit>
        <low>0.0</low>
        <high>10.0</high>
      </displayLimit>
      <linearConvert>
        <engUnitsLow>0.0</engUnitsLow>
        <engUnitsHigh>10.0</engUnitsHigh>
      </linearConvert>
    </currentInput>
  </powerSupply>
</record>
```

PowerSupplyRecord instance

```
<voltageInput>
  <aiRaw>
    <input linkSupportName = "asynInt" configStructureName = "asynLink">
      <portName>somePort</portName>
      <addr>2</addr>
    </input>
  </aiRaw>
  <units>volts</units>
  <displayLimit>
    <low>0.0</low>
    <high>2.0</high>
  </displayLimit>
  <linearConvert>
    <engUnitsLow>0.0</engUnitsLow>
    <engUnitsHigh>2.0</engUnitsHigh>
  </linearConvert>
</voltageInput>
</powerSupply>
<priority>medium</priority>
</record>
```

PV Naming

- EPICS V3 – pvname is <record>.field
- javaIOC - <record>.name.name. ...
 - Assume “<record>.” thus name.name. ...
- Why “name” instead of “field” ?
 - Could be field, i.e. Traverse structure hierarchy
 - Properties provides better way

Property

- Assigned to field, structure, recordType
- Provides two features
 - When assigned to a field
 - References a field that has associated information
 - When assigned to a structure or recordType
 - Look elsewhere for the data

Field Property

```
<structure name = "aiRaw">  
  <field name = "value" type = "int" >  
    <property name = "status" associatedField = "status" />  
    <property name = "severity" associatedField = "severity" />  
    <property name = "timeStamp" associatedField = "/timeStamp" />  
  </field>  
  <field name = "status" type = "string" />  
  <field name = "severity" type = "menu" menuName = "menuAlarmSevr" />  
</structure>
```

Field Property Comments

- The field value has properties:
 - status: the data is in aiRaw.status
 - severity: the data is in aiRaw.severity
 - timeStamp: start looking at top of record for property or field timeStamp.

Structure Property

```
<structure name = "aiLinear">  
  <property name = "rawValue" associatedField = "aiRaw.value" />  
  <field name = "aiRaw" type = "structure" structureName = "aiRaw"/>  
</structure>
```

Structure Property Comments

- structure aiLinear has property rawValue
 - Look in aiLinear.aiRaw
 - In aiRaw look for value
 - In this example that is the location of value
 - The search can be recursive
 - The search can lead another record
- A recordType is also a structure
 - Thus it can have properties

Locating fields in a powerSupplyRecord

- The following slide shows the actual field and associated property fields for the power, current, and current.rawValue
- Again “<recordName>.” is left off the pvname

pvnames for a powerSupplyRecord

power => powerSupplyRecord.powerSupply.power

properties are:

status => powerSupplyRecord.powerSupply.status

severity => powerSupplyRecord.powerSupply.severity

units => powerSupplyRecord.powerSupply.units

displayLimit => powerSupplyRecord.powerSupply.displayLimit

timeStamp => powerSupplyRecord.timeStamp

current => powerSupplyRecord.powerSupply.currentInput.value

properties are:

status => powerSupplyRecord.powerSupply.currentInput.status

severity => powerSupplyRecord.powerSupply.currentInput.severity

units => powerSupplyRecord.powerSupply.currentInput.units

displayLimit => powerSupplyRecord.powerSupply.currentInput.displayLimit

timeStamp => powerSupplyRecord.timeStamp

current.rawValue => powerSupplyRecord.powerSupply.currentInput.aiRaw.value

properties are:

status => powerSupplyRecord.powerSupply.currentInput.aiRaw.status

severity => powerSupplyRecord.powerSupply.currentInput.aiRaw.severity

timeStamp => powerSupplyRecord.timeStamp