

EPICS Meeting 2014, CEA, October 2014

# New Features in EPICS V4 Release 4.4

Marty Kraimer, Matej Sekoranja

Your **TRUSTED** Control System Partner





## □ pvData types:

### *scalar*

boolean int8 int16 int32 int64 uint8 uint16 uint32 uint64  
float double string

### *scalarArray*

one dim array of any scalar type

### *structure*

each field can be any type including structure

### *structure array*

array of structure fields with each having same  
introspection interface

### ***union !!!!! new type***

single sub-field which can change type dynamically

### ***unionArray !!!!! new type***

array of union fields

## □ union has two flavors

### ■ variant

- The field can have any PVData type

### ■ regular union

- The set of allowed types can be specified

```
structure NTVariantArray
  int dataType
  boolean [] booleanValue
  byte [] byteValue
  short [] shortValue
  int [] intValue
  long [] longValue
  float [] floatValue
  double [] doubleValue
  string [] stringValue
```



```
structure NTUnion
  union value
  boolean [] booleanValue
  byte [] byteValue
  short [] shortValue
  int [] intValue
  long [] longValue
  float [] floatValue
  double [] doubleValue
  string [] stringValue
```

```
structure NTVariantArray
  int dataType
  boolean [] booleanValue [false,true,false]
  byte [] byteValue []
  short [] shortValue []
  int [] intValue []
  long [] longValue []
  float [] floatValue []
  double [] doubleValue []
  string [] stringValue []
```



```
structure NTUnion
  union value
    boolean [] booleanValue [false,true,false]
```

## pvData – new features



- ❑ pvDataCPP enforces Copy on Write for Arrays
  - This greatly helps with developing support for multiCore architectures.
  
- ❑ FieldBuilder
  - Easy way to create introspection objects
  
- ❑ Support for fixed size and bounded size arrays.

- ❑ Revised pvAccess API
  - ❑ Minor, but backward in-compatible changes, to make API more user-friendly
- ❑ Pluggable pvAccess security implementation support
  - ❑ e.g. pvaSrv implements CA asLib security plugin
- ❑ “codec” based transport
  - ❑ This means: protocol logic only in one place for multiple transports (TCP, UDP, ..., shmemp, etc.)
  - ❑ Well tested and stable code (lots of regression tests)
- ❑ Local multicast feature removes the need for “caRepeater”-like process
- ❑ *pvlist* tool (find running servers, get server info and list of channels)
- ❑ “wildcard” services (allows channel names like “\*:twiss”)

- ❑ *normativeTypes* defines a set of standard structures. The purpose is to have well known structures for use by services and so that tools can be created to make it easy to use the standard structures.
  
- ❑ Summary of existing status:
  - Only C++ has helper classes
  - *normativeTypesCPP* has major changes
  - Builders make it easy and safe to create an instance.
  - Support for alarm, timeStamp, control, display, enum
  - Implements NTScalar, NTScalarArray, NTNameValue, NTTable, NTMultiChannel, NTNDarray
  - More in future releases

## □ NTScalar

- A scalar value field and optional fields like alarm, timeStamp, etc. Like the data available from any of the iocCore records that have a scalar VAL field.

## □ NTScalarArray

- A scalarArray value field an optional fields. Like the data available from any of the iocCore records that have an array VAL field.

## □ NTNameValue

- Just what the name implies, i. e. name,value pairs. A primary use is client arguments for a service.

## □ NTTable

- The value field has a set of subfields which are each a scalar array. The set of subfields is the rows of the table and each subfield is a column.

## □ NTMultiChannel

- Used for data from a set of channels. Used to get/restore machine state, archiving, and control.

## □ NTNDarray

- A C++ wrapper for areaDetector.

# pvlist example



```
mrk> pvlist
GUID 0x2A32BF6FF57E443C835F8336, version 1: tcp@[10.0.0.3:48028]
GUID 0x526FDACC23314A7D85CBB097, version 1: tcp@[10.0.0.3:5075]
GUID 0x9D1FBC9F41FD47399CCC3731, version 1: tcp@[10.0.0.3:50974]

mrk> pvlist 10.0.0.3:48028
powerSupplyArray
scalarDouble
dacCurrentEmbedded
psSimple
event
counterInput
current
dacCurrentSupported
counterMonitor

mrk> pvlist -i 0x9D1FBC9F41FD47399CCC3731
structure
string process 8433
string startTime 2014-10-17T10:11:03.285
string version 4.0.0
string implLang java
string host atlas
string os Linux 3.2.0-4-amd64
string arch amd64
int CPUs 8
```

## Compatibility Issues



- ❑ The changes to the pvAccess API, Copy On Write, and other changes mean that all code that uses pvAccess and C++ code that uses arrays must be changed. We know that starting with V4 release 4.4 compatibility must be an important issue when future changes are considered.
- ❑ The changes were made knowing that it was not compatible.
- ❑ The changes were the results of discussions by the V4 developers over the last three or so years.

# Status

EPICS V4 release 4.4.0\_pre1 Imminent

EPICS V4 release 4.4.0 Soon

MS Visual Studio Support

Peter Heestermann

EPICS V3, Asyn, pvData/pvAccess,etc

**Thank you!**

Your **TRUSTED** Control System Partner

