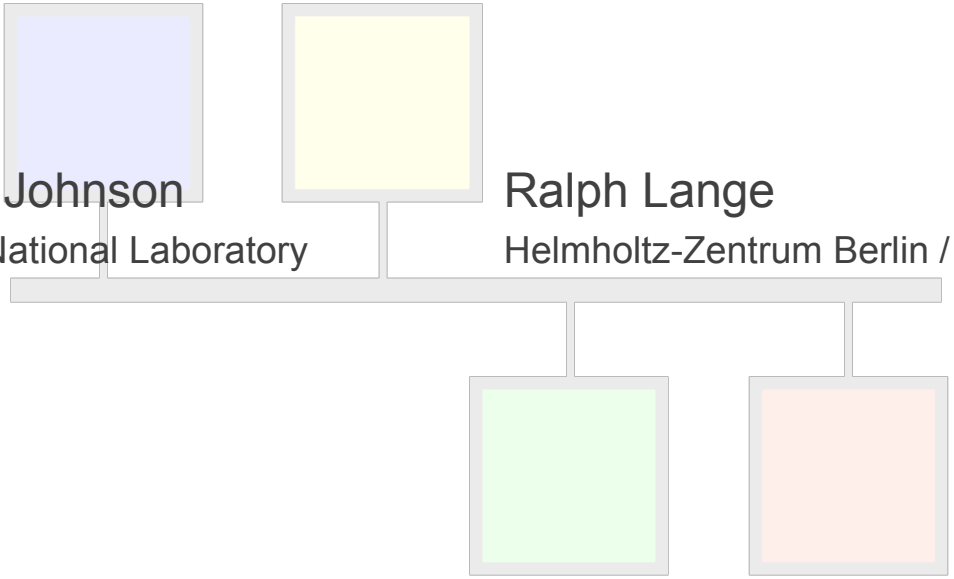
Argonne NATIONAL LABORATORY

V2

Evolutionary Plans for EPICS Version 3

Andrew Johnson
Argonne National Laboratory

Ralph Lange
Helmholtz-Zentrum Berlin / BESSY-II



U.S. DEPARTMENT OF ENERGY

Outline

- EPICS
- Evolution
- Process Variable Names
- Record Aliases
- Field Modifiers
- JavaScript Object Notation (JSON)
- JSON Data Encoding
- JSON Field Modifiers

EPICS

- 23% of abstracts use the keyword “EPICS”
  - Real number of related submissions is higher
- EPICS has a significant penetration
  - Important at many sites
- Most EPICS users are large, long-term projects
  - Run EPICS “Version 3”, Base 3.13.x or 3.14.x
- Requirements for large sites
  - Transparent communication between different versions
  - Migrate to new versions without making major application changes
- How many Channel Access client programs are there?
  - Some are no longer actively maintained
  - Not easily modified to use extensions to the CA API

How Evolution Benefits EPICS Users

- New versions are more successful if
  - They can be introduced piecemeal
    - "All or nothing" conversions very hard to achieve on large projects
  - New features can be introduced gradually
    - Systems that need them can be upgraded first
  - Existing applications (IOCs and clients) still build and run
    - Requiring minor changes is acceptable
- Desirable characteristics:
  - Compile-time indication of any necessary application code changes
    - Better to fail when compiling than when in operation
  - Provide conversion tools for significant modifications
  - New server features can be used by older version clients

Process Variable Names

- IOC PV names use this syntax:
  - *record name* — Uses **VAL** field
  - *record name* **.** — Uses **VAL** field
  - *record name* **.** *field name*
- A space character ends the PV name
  - Requirement of the PV link field parser in the IOC
- *record name* part ends at first period character
  - Documented restrictions on the *record name* not strictly enforced
  - Too late to start enforcing other limitations now
- *field name* part is a name defined by the record type
  - Valid C identifier
    - All alphanumeric or underscore
    - Starts with letter or underscore
  - Limitations enforced by compiler when building record type

Record Name Aliases

- Sites discover problems in naming standard after machine in operation
- Renaming records requires updating all CA clients that know the old names
  - A tool like IRMIS can help find many of the old references
  - This is an “all or nothing” change, often not simple

- EPICS Base 3.14.11 introduced record aliases
  - Additional names for a record
  - Configured in the record instance definition, or defined separately
  - A CA client can discover the canonical name using the .NAME field
- Aliases permit record renames to take extended periods of time
  - Both names work, so CA clients can be updated at leisure
  - An “alias watcher” CAS tool could be written to monitor CA searches and report client machines still using an old name

Field Modifiers

- Limitations of PV *field name* characters guarantees no others in use
- Extensions to the PV name syntax can be made after the dot

- Base R3.14.11 accepts PV names with this syntax
  - *record name* **.$** Uses **VAL** field
  - *record name* **.** *field name* **$**
- The dollar sign is a field modifier
  - modifies the meta-data for the field, or other aspects of the channel
- The dollar field modifier is valid on string or link field types
  - Changes the native data type reported to “array of DBF_CHAR”
  - Allows existing CA clients to transport strings longer than 40 characters
    - Both MEDM and EDM support using character arrays as text strings

Future Field Modifiers

- This syntax has been implemented experimentally
  - *record name* `.[` *array offset* `]`
  - *record name* `.` *field name* `[` *array offset* `]`
- Where *array offset* is an integer
- Allows access to contiguous subset of elements of an array field
  - Currently array accesses all start from the first element

- Other modifiers are possible, e.g.
  - *record name* `.` *field 1* `,` *field 2* `,` *field 3* ...
  - Return an array atomically constructed from the listed fields
  - What CA data-type should this use?

JavaScript Object Notation (JSON)

- "JavaScript Object Notation (JSON) is a light-weight, text-based, language-independent data interchange format. … JSON defines a small set of formatting rules for the portable representation of structured data."
- Defined as an Internet standard, RFC 4627
- Support libraries available for all common computer languages
- Visit http://www.json.org/ for details, including syntax charts

- Easy to read and create by hand, e.g.
  - `{"RA":[18,53,35.079], "DEC":[33,1,45.03], "Name":"M57"}`

- EPICS Base 3.15 will include a copy of the YAJL support library (C/C++)
  - Small, event-driven parser, minimizes memory allocations

JSON For Data Encoding

- Returning to the list field modifier example
  - *record name* `.` *field 1* `,` *field 2* `,` *field 3 ...*
- CA get or monitor could return a JSON-encoded list of values
  - `[` *value 1* `,` *value 2* `,` *value 3 ...* `]`
- Can mix string and numeric data values

- New record types could support JSON-encoding in string fields
  - Allows RPC behavior, e.g.
  - ```
    caput -S WHT:TCS:CMD.$ \
        '"SLEW":{"RA":[18,53,35.079],"DEC":[33,1,45.03]}'
    ```
    ```

JSON For Field Modifiers

- Can use JSON to encode complex field modifiers
  - *record name* **.{** *JSON object* **}**
  - *record name* **.** *field name* **{** *JSON object* **}**
- The *JSON object* contains one or more name/value pairs
  - The name selects a particular plug-in for this channel
  - The value can be a single configuration parameter, a JSON list, or a nested JSON object specifying multiple parameters

- Example from Ralph Lange's poster paper THP090
  - **`Current.VAL{"rate":0.5}`**
    - Using the "rate" plug-in, limit the event update rate of this channel to at most 0.5Hz
  - **`Current.VAL{"rate":{"min":0.1,"max":0.5}}`**
    - Using the "rate" plug-in, limit updates over this channel to between 0.1 and 0.5Hz
  - The "rate" plug-in above could be written to handle both kinds of parameter

Conclusions

- EPICS Version 3 can be extended without breaking compatibility with existing client and IOC applications
- Record aliases available in EIPCS Base 3.14.11
- Field modifiers are coming
  - The long string modifier can be used with 3.14.11 IOCs
  - Additional modifiers will be added in subsequent releases.
- JSON is an extremely promising standard
  - Will be used for extensible field modifiers in 3.15 IOCs
  - May have many additional uses in the future