APS-U Component Database eTraveler

Ned Arnold / Dariusz Jarosz / Sinisa Veseli
Argonne National Laboratory

EPICS Collaboration Meeting @ APS
June 15, 2018
The Need ...

• Do we have all the parts?
• Where are they?
• Are they ready?
• Are you sure?
The Need ...

Integrated Relational Model of Installed Systems

Web Desktop
- AC Power Info
- AOI Info
- Component Type Info
- IOC Info
- Network Info
- PLC Info
- Racks Info
- Server Info
- Spares Info
- Component History
- Web Desktop Help

IRMIS Desktop (idt)
- idt: PV Info
- idt: IOC
- idt: Component
- idt: Component-Type
- idt: Cables
- idt: Help

Global Search Tool
Search

Database Server:
ctrdbprod:irmis
Application Server:
ctlappsirmis
Component Database (CDB)

What Does It Do?

- **Assists staff** in tracking components
  - Provides a mechanism to associate drawings, documents, specifications, requisitions, … to a component
  - Define additional properties that a set of components have in common
  - Allows for identifying and tracking inventory items (where installed, repair record, periodic maintenance, …)
  - Captures installed components (type / instance / location)

- **Allows for “Project-wide” processes** to be applied
  - Common naming conventions (good luck)
  - QA Properties and work flow
  - Organizing of eTravelers
What is a "component"?
- Something you design
- Something you buy
- Something you build/assemble
- Something you refer to … even if it doesn’t exist

What do you call it?
- Part / assembly / widget / …
- Component / component-type / component instance / …
- Configuration item / lattice element / accelerator component / …
Component Database (CDB)

Vocabulary

**Catalog**
(each unique type of component or component design or COTS item + properties/drawings/specification/..)

**Inventory**
(each unique instance of component procured or fabricated) + properties/serial #/QR code/travelers/pictures/…)

**Machine Designs**
(A group of catalog components to perform a particular function + inventory items to build it + Properties/pictures/locations/…)

N. Arnold – EPICS Collaboration Meeting – June 2018
Component Database (CDB)

WARNING!!!! THREE DIFFERENT NAME SPACES!!!!

Vocabulary

Catalog Item
DMM Quadrupole

Machine Design

S27A:Q1

Machine Design Item is a place holder or "address". It has …
- A reference to a Catalog Item
- A reference to an Inventory Item
- It’s own properties, history, etc.

Inventory Items:
Specific units of “DMM Quadrupole” can be tracked for inspection, testing, where installed, maintenance log, etc.

Q1 Production - [Unit: 1]
Q1 Production - [Unit: 2]
Q1 Production - [Unit: 3]
Component Database (CDB)

WARNING!!!! THREE DIFFERENT NAME SPACES!!!!

Vocabulary

Catalog Item
Libera Brilliance+ (APS Model)

Inventory Items:
Specific units of “Libera Brilliance+ (APS Model)” can be tracked for inspection, testing, where installed, maintenance, etc.

Machine Design Item is a place holder or “address”. It has ...
- A reference to a Catalog Item
- A reference to an Inventory Item
- It’s own properties, history, etc.

3/15/2022: Installed Libera+/APS Unit 14 into S27A:LIB1

3/16/2022: Installed Libera+/APS Unit 19 into S28A:LIB1

N. Arnold – EPICS Collaboration Meeting – June 2018
Catalog Entry

Optional Properties

Common Properties
Inventory Entry

APS / CDB
000 000 147
Browse using Common Properties

Links to Catalog Items

Links to Inventory Items

<table>
<thead>
<tr>
<th>Acry</th>
<th>Primary Image</th>
<th>Name</th>
<th>Model Number</th>
<th>Alternate Name</th>
<th>Description</th>
<th>Inventory</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>DMM Top Level Design (E1)</td>
<td></td>
<td></td>
<td>Top level design of the Dummy Modular Multipllet prototype</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>DMM MAGNET SUPPORT PLATE U22102202-110001</td>
<td></td>
<td></td>
<td>SUPPORT PLATE FOR DMM MAGNETS, INTERFACE TO DMM SUPPORTS AND VACUUM COMPONENTS</td>
<td>DMM-SUPPORT-PLATE-A001</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DMM QUADRUPOLE COIL ASM U22102202-104000</td>
<td></td>
<td></td>
<td>DMM QUADRUPOLE COIL</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>DMM QUADRUPOLE U22102202-111100</td>
<td></td>
<td></td>
<td>QUADRUPOLE ASSEMBLY, STEEL POLES</td>
<td>DMM-QUADRUPOLE-A001</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DMM QUADRUPOLE U22102202-111000</td>
<td></td>
<td></td>
<td>QUADRUPOLE ASSEMBLY, STEEL POLES, EDM, SYNCHROTRON VACUUM CHAMBER CUTOUT</td>
<td>DMM-QUADRUPOLE-A002</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DMM QUADRUPOLE U22102202-113100</td>
<td></td>
<td></td>
<td>QUADRUPOLE ASSEMBLY, STEEL MUSHROOM POLES, EDM</td>
<td>DMM-QUADRUPOLE-A003</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DMM QUADRUPOLE U22102202-114100</td>
<td></td>
<td></td>
<td>QUADRUPOLE ASSEMBLY, VP POLES</td>
<td>DMM-QUADRUPOLE-A004</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DMM Sextupole Coil</td>
<td></td>
<td></td>
<td>Sextupole Coil Data</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>DMM Sextupole S1A001</td>
<td></td>
<td></td>
<td>LATTICE V3 Sextupole S1 Design</td>
<td>S1A001</td>
</tr>
<tr>
<td></td>
<td></td>
<td>M1 Dipole pre-prototype</td>
<td></td>
<td></td>
<td>First prototype magnet (from Fermilab)</td>
<td>First Unit (from Fermilab)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>M4 Potting Mold Assembly u22102202-112300</td>
<td></td>
<td></td>
<td>M4 Potting Mold Assembly</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>M4 Trim CoiWinding Fixture Assembly u22102202-112200</td>
<td></td>
<td></td>
<td>M4 Trim CoiWinding Fixture Assembly</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>M4 Winding Fixture Assembly u22102202-112100</td>
<td></td>
<td></td>
<td>M4 Winding Fixture Assembly</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Prototype 8-Pole Corrector</td>
<td></td>
<td></td>
<td>Prototype 8-Pole Fast Corrector built by BNL</td>
<td>Unit: 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Prototype M4 Transverse Gradient Quadrupole U22102202-200000</td>
<td></td>
<td></td>
<td>Prototype M4 Transverse Gradient Quadrupole</td>
<td>M4-PROTOTYPE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Prototype Q8 Quadrupole U22102202-130000</td>
<td></td>
<td></td>
<td>Prototype Q8 Quadrupole</td>
<td>Unit: 1, Unit: 2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Quadrupole 0.213m</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Quadrupole 0.259m</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Quadrupole 0.463m</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Quadrupole 0.688m</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

N. Arnold – EPICS Collaboration Meeting – June 2018
Features …

- Numerous view features can be customized for an individual or a group
  - Item filters
  - Columns displayed
  - Rows per page
  - Favorites / Owned / …
Features ...

- **User Interfaces**
  - Web Portal
  - Customized views for individuals and groups
  - REST Web Service (Python and Java APIs)
  - Command Line Interfaces (built on top of Python APIs)
  - Mobile Apps (in development)

- **Privileges**
  - Authentication: APS LDAP
  - Authorization: Users are members of one or more CDB-defined groups. You can modify entities you own and entities owned by your CDB-defined group if they are set to “group writable”.
  - Administrators: Certain tables can only be modified by Users assigned to the CDB_ADMIN group
### Properties

<table>
<thead>
<tr>
<th>Catalog Items</th>
<th>Inventory Items</th>
<th>Machine Design Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name</td>
<td>Name</td>
</tr>
<tr>
<td>Model Number</td>
<td>Tag</td>
<td>Machine Tag</td>
</tr>
<tr>
<td>Alternate Name</td>
<td>Catalog Item (automatic)</td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>Description</td>
<td>Description</td>
</tr>
<tr>
<td>Project</td>
<td>Project</td>
<td>Project</td>
</tr>
<tr>
<td>Technical System</td>
<td>QR #</td>
<td></td>
</tr>
<tr>
<td>Function</td>
<td>Serial Number</td>
<td></td>
</tr>
<tr>
<td>Sources (Vendors)</td>
<td>Location</td>
<td></td>
</tr>
<tr>
<td>Location</td>
<td>Location Details</td>
<td>Location Details</td>
</tr>
<tr>
<td>Location Details</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<< Optional Properties >>
- Links to Drawings & Documents
  - Images
  - Web Links
- Uploaded Documents
  - Form Factor
  - eTraveler Templates
  - QA Level
  - QA Requirements
  - Link to Software Support Modules

<< Optional Properties >>
- Purchasing Info
  - Images
  - Uploaded Documents
  - Revision #
  - eTraveler Instances
  - Inspection Results
  - Test/Measurement Results

<< Optional Properties >>
- IP #
- Host Name
- GPIB Address
- VME/VXI Configuration
- Location in Rack
- Radiation Safety Component
- Link to Source Code

N. Arnold – EPICS Collaboration Meeting – June 2018
Features ...

- **Catalog Item Classifications**
  - Classifications allow catalog components to be grouped by technical system, function, or ownership. If categorized carefully, items can be found expediently.
  
  - **Function**: generic function(s) that the component performs (e.g. stand, magnet, ADC, vacuum chamber, etc).
  
  - **Technical System**: Items are further grouped into “Technical Systems” which are intended to group component types by discipline (e.g. Diagnostics, Controls, Accelerator, Beamline, RF, ...).
Features ...

- Properties
  - Properties provide a flexible mechanism for capturing object-dependent information (e.g. not all components will have the same metadata)
  - Property types may be associated with a restrictive set of “allowed” property values
  - Property types may be linked to unique a “handler”, which adds specific functionality to that property
    - ICMS Handler, PARIS Handler, PDMLink Handler, …
  - A time-stamped history of each property value is kept to provide a historical log
Features ...

- **Assemblies**
  - A simple hierarchy of components can be created
  - Discussion Starter: Should you create “assemblies” in the Catalog or in the Machine Design?

- **Mini-logbooks**
  - Chronological log entries for each item *in all domains*
  - Upload an attachment to a log entry
Features ...

Component Database System Components

PTC Windchill
(Mechanical Dwgs)

Document Management System

FRIB eTraveler

REST Web Service

Document Repository

System Tools

Database

WWW

ICMS

EDP

PARIS

AMOS

Web Browser

Mobile Apps

CLI Tools

User Scripts

CDB Component

External Application

External Client Tools

User Authentication

Planned Additions

N. Arnold – EPICS Collaboration Meeting – June 2018
Features ...

- Integration with Mechanical Drawing Repository
Features ...

- Integration with FRIB’s Traveler Application

---

**FRIB Traveler v3.0**
A Web application to design, carry out and organize processes

_You might find the previous version document at [github](https://github) is still helpful before I have all the sections finished in this document._

---

**How to use this document**
There is an Audience statement on the top of each section. If you are not the target audience, then you can skip the section.

**Basics of the traveler application**
**Audience:** all users

The traveler application is a Web application for design, carry out, and organize electrical process documents, which we call travelers. It provides a Web interface for edit and manage forms. Furthermore, users can organize travelers by binder. The application provides a limited HTTP API to read the traveler information.

**What is a traveler?**
A traveler is an electrical document that is designed to support the execution of a predefined process and to collect user input data and notes in the process. A typical traveler user case is to implement a work instruction that specifies all the steps to accomplish a work.

A traveler has properties like title, description, deadline, locations, and tags. The user can add/remove a tag into the tag list. The tag can be a device name defined in CCDB or any string. A traveler is initialized when it is created. Its state can be changed to active, submitted for completion, completed, and frozen. A traveler can be archived. Only the traveler owner can access the traveler when it is archived. A traveler owner can share her/his traveler with other users/groups. A user can also transfer the ownership of a traveler to other user.

The process and inside user inputs are defined in a form. The users with written permission can input values into an active traveler. The input history is kept in the traveler, and shown under each input. Each input can also have user notes attached to it. A traveler can be considered as the composition of a form, the input data, and the notes.

`traveler = form + data + notes`

The [travelers section](https://argonne.gov/traveler) provides more detailed information about how to use and manage travelers.
Features ...

- **Integration with FRIB’s eTraveler Application**

**Catalog**
(each unique type of component or component design or COTS item + properties/drawings/specification/…)

**Inventory**
(each unique instance of component procured or fabricated) + properties/serial #/QR code/travelers/pictures/…

**Machine Designs**
(A group of catalog components to perform a particular function + inventory items to build it + Properties/pictures/locations/…)

**eTraveler Templates/Forms**
(an electronic form designed to guide the user through a set of steps <for specific component types>)

**eTraveler [Instances]**
(A copy of a Traveler Template filled in for a particular instance of a part)

- Hydro Test for Mask Type #2: Qrid = 000 001 679
- Hydro Test for Mask Type #2: Qrid = 000 001 682
- Hydro Test for Mask Type #2: Qrid = 000 001 683
- Vacuum Test Type #2
- ACL for Mask Type #2
- Vacuum Certification

*eTraveler Example*
Features ...

- Integration with FRIB’s Traveler Application

Catalog Items have “Templates (Forms)”
Features ...

- Integration with FRIB’s Traveler Application

Inventory Items have "Instances"
Features ...

- Integration with FRIB’s Traveler Application
  - Supports “binders” (groups of eTravelers)
  - Traveler specific columns with info fetched from traveler system
Features ...

- Coming Soon ...
  - Machine Design Domain
  - User interface will be very tricky ...
    would like it to be graphical
  - Capture an exhaustive Bill of Materials for the entire accelerator??
Features ...

- Coming Soon ...
  - MAARC – **Measurement and Analysis Archive**
  - User Interface to “Data Management” entries

- Data uploaded via Data Management scripts
- Captures user-defined meta-data for each item
- Items can be associated with any other CDB item
- First use will be Magnet Measurement data
Features ...

- Coming Soon ...

  - Mobile Apps

    Computer / Tablet / Phone
    ... can run the CDB application in any browser

    Mobile computer
    ... with wifi and scanner that runs custom applications
    (e.g. APS Stockroom)

    Handheld Scanners
    (bluetooth to tablet with custom tablet application)

    • Takes photos
    • Scans docs to pdf

    < a URL is encoded in the symbol which directs a browser to the CDB page for that QrId >

    < these tags require a “CDB-aware” application that knows what to do with the number (encoding a URL would be too long) >

N. Arnold – EPICS Collaboration Meeting – June 2018
Technologies

- **CDB**
  - MySQL (DB)
  - Java / Glassfish (Web Portal)
  - PrimeFaces (UI framework)
  - CherryPy (Web services)

- **Future Needs ...**
  - Generate reports from data in eTravelers
    - e.g. “Display the width measurement for each bellows received”
  - Item Relationships (like IRMIS 2.0, circa 2005)
  - Cable Management
  - Directory Service / Name Database
Install ... (djarosz@anl.gov)

- **Prerequisites command for red-hat is:**
  - yum install -y gcc libgcc expect zlib-devel openssl-devel openldap-devel readline-devel git make cmake sed gawk autoconf automake wget mysql mysql-libs mysql-server mysql-devel curl

- **Prerequisites command for debian is:**
  - apt-get install wget gcc git make cmake build-essential libcurses-ocaml-dev curl expect mysql-server libmysqlclient-dev openssl libssl-dev libldap2-dev libsasl2-dev sed gawk

**Deployment Process:**

- --- Create a directory for installation and cd into it
- mkdir install
- cd install
- --- Get the repository from github [https://github.com/AdvancedPhotonSource/ComponentDB](https://github.com/AdvancedPhotonSource/ComponentDB)
- --- NOTE: we also have releases on the page for more stable code)
- git clone https://github.com/AdvancedPhotonSource/ComponentDB.git
- cd ComponentDB
- --- Get additional support installed within the cdb install directory
- make support
- source setup.sh #load env with new support
- --- Follow on screen instructions to generate required configurations for the application
- make configuration
- --- Add necessary data to mysql database
- make clean-db
- --- Apply configuration and configure connections to db for glassfish
- make configure-web-portal
- --- finally deploy the portal
- make deploy-web-portal