

Quick overview of Day 1

- There were some questions during Day 1 sessions that were either unanswered or not answered clearly.
- Also there have been some updates since then.

We like to go through them now...

First, changing the 'curl' usage...

- After Day 1, it was pointed out that the suggested curl line did not work for some users.

Original:

```
curl --cert-type P12 --cert MyCert.p12:myPSWD 'https://dbwebapi2.fnal.gov:8443/cdbdev/api/version/...'
```

- This is due to a different version of OpenSSL library. See [here](#) for more details.

We have updated the method so that it works, in dependent of your OpenSSL ver.

```
curl --cert Output.pem --pass YourPhrase 'https://dbwebapi2.fnal.gov:8443/cdbdev/api/version/...'
```

Where do Output.pem and YourPhrase come from?

Extracting your pem with a phrase

- By now, you must have download your p12 file from CILogon.

You can convert this to the pem format (Privacy Enhanced Mail, another container format).

- In your terminal, do this:

```
openssl version
```

If it returns “OpenSSL 3.X.X” or newer, then do this:

```
openssl pkcs12 -in uercred.p12 -out Output.pem --legacy
```

where `uercred.p12` is your downloaded p12 file and

`Output.pem` is your converted pem file. Give a filename you like!

Else (if your OpenSSL is older than v3), then do this:

```
openssl pkcs12 -in uercred.p12 -out Output.pem
```

- While performing the above conversion, you will be asked the followings:

- ▶ “Enter Import Password” : Provide your P12 password.
- ▶ “Enter PEM passphrase” : Give a unique phrase. **Remember this!!**
- ▶ “Verifying - Enter PEM pass phrase” : Enter the above phrase again.

From the last week, let's do this together!

Post a single item

- Get "PostItem.json" file from the indico site.
It should look like this:

```
{
  "component_type": {
    "part_type_id": "Z00599800001"
  },
  "country_code": "US",
  "comments": "Testing...",
  "institution": {
    "id": 186
  },
  "manufacturer": {
    "id": 7
  },
  "specifications": {
    "DATA": {
      "Last name": "Muramatsu",
      "First name": "Hajime"
    }
  }
}
```

- Then execute the command line below:

```
CURL -H "Content-Type: application/json" -X POST -d @PostItem.json "${APIPATH}/component-types/Z00599800001/components" | jq
```

Were you able to post a new item?

Can you see it with the WEB-UI?

A (almost) empty Item Specifications def.

Hajime Muramatsu U of Minnesota DUNE HWDB Tutorial JUL/15/2024 49

Datasheet: example 2

Datasheet

```
AR: []
Gain:
- T1
- T2
Size: null
Type: null
```

- One could define an array ([]). In this example, it is an empty one.
You can put anything in there later.
- Also, one could define Options, such as:
 - Gain:**
 - T1
 - T2(could also type as "Gain: [T1,T2]")

- Last week, I said one could define with an empty array.

Let me go into this with a bit more explanation today...

- Let's look at the definition of the Component Type that we just posted to: **Z00599800001**

A (almost) empty Item Specifications def.

Edit Component Type

[SPECS LOG](#) [IMAGES](#)

Type Name Test Type 001

Type ID 00001

Full Name Z.FD1-HD HVS.HWDBUnitTest.Test Type 001

Part Type ID Z00599800001

Comments

Category generic

Managed by x tester

Manufacturers x CERN x Hajime Inc

Created 2024-07-19 11:44:51

Created by Hajime Muramatsu

Specifications

Version	1
Created	2024-07-19 11:44:51-05:00
Created by	Hajime Muramatsu
Datasheet	DATA: {}

Connectors

select a type

- Not an empty array,
but we have an empty blob in this particular example.

A (almost) empty Item Specifications def.

- Change something inside of your "DATA".

- Try to post it again:

```
{
  "component_type": {
    "part_type_id": "Z00599800001"
  },
  "country_code": "US",
  "comments": "Testing...",
  "institution": {
    "id": 186
  },
  "manufacturer": {
    "id": 7
  },
  "specifications": {
    "DATA": {
      "Last real name": "Smith",
      "First real name": "David"
    }
  }
}
```

```
CURL -H "Content-Type: application/json" -X POST -d @PostItem.json "${APIPATH}/component-
types/Z00599800001/components" | jq
```

- ... or even something like this...

```
{
  "component_type": {
    "part_type_id": "Z00599800001"
  },
  "country_code": "US",
  "comments": "Testing...",
  "institution": {
    "id": 186
  },
  "manufacturer": {
    "id": 7
  },
  "specifications": {
    "DATA": {
      "Public Names" : [
        {
          "Last name" : "Muramatsu",
          "First name" : "Hajime"
        },
        {
          "Last name" : "Petrov",
          "First name" : "Andrey"
        }
      ],
      "Real Name" : {
        "Last real name" : "Smith",
        "First real name" : "David"
      }
    }
  }
}
```

- When you define with an empty blob/array, within the empty blob/array, it will accept any key-name/structure as long as they are in the accordance with YAML.
- This feature might be particularly useful when one needs to transfer data from MongoDB.

Characters not allowed in the Type sheets

	A	B	C	D	E	F	G
1	Project	System Name	System ID	Subsystem Name	Subsystem ID	Type Name	Type ID
2							
3	D	FD1-HD HVS	5	HWDBUnitTest	998	Test Type 012	12
4	D	FD1-HD HVS	5	HWDBUnitTest	998	Test Type 013	13
5	D	FD1-HD HVS	5	HWDBUnitTest	998	Test Type 014	14
6	D	FD1-HD HVS	5	HWDBUnitTest	998	Test Type 015	15
7	D	FD1-HD HVS	5	HWDBUnitTest	998	Test Type 016	16
8	D	FD1-HD HVS	5	HWDBUnitTest	998	Test Type 017	17
9	D	FD1-HD HVS	5	HWDBUnitTest	998	Test Type 018	18
10	D	FD1-HD HVS	5	HWDBUnitTest	998	Test Type 019	19
11	D	FD1-HD HVS	5	HWDBUnitTest	998	Test Type 020	20

Cautions: For names in the sheet, you cannot use the following characters:
comma, period, quotation marks, underscore or the percent sign.

- We have updated the Training site and the corresponding talk slide.

Diff. in WEB-UI between dev. and pro. ?

Development

DUNE Hardware DB
DEEP UNDERGROUND NEUTRINO EXPERIMENT

Home
Component Types
Items
Institutions
Images
Manufacturers
Users
Admin <
Logout
Requests Issues?

Overview

The components database tracks parts and what other, if any, part is attached to it. The database is designed around physicists describing what each part is and how those parts are fit together. First one must create a "type" which defines what a part is and the data collected for it. Then the data is entered for a specific "type". Tests may also be defined for a specific product "type", and data entered for each physical piece. Each physical part tracked must have some external id, which is unique across the experiment, entered with that part.

REST API

A REST API is available for retrieving and adding data to the database. Using the API requires a certificate. It also requires that you be recognized as a member of DUNE. Be aware that FERMI only accepts CILogon certificates.

Online Documentation: [Swagger](#)[Redoc](#)
[REST Tutorial](#)

Vincit qui se vincit.

He conquers who conquers himself.

Database Schema

This product is created over a PostgreSQL database. Two views of the schema are available. One which shows all the tables. There is also a second simpler view of the schema which does not show the security tables and their connections.

[Full Schema](#)
[Simplified Schema](#)
[Database Conceptual Overview](#)

Production

DUNE Hardware DB
DEEP UNDERGROUND NEUTRINO EXPERIMENT

Home
Component Types
Items
Institutions
Images
Manufacturers
Users
Admin <
Logout
Requests Issues?

Overview

The components database tracks parts and what other, if any, part is attached to it. The database is designed around physicists describing what each part is and how those parts are fit together. First one must create a "type" which defines what a part is and the data collected for it. Then the data is entered for a specific "type". Tests may also be defined for a specific product "type", and data entered for each physical piece. Each physical part tracked must have some external id, which is unique across the experiment, entered with that part.

REST API

A REST API is available for retrieving and adding data to the database. Using the API requires a certificate. It also requires that you be recognized as a member of DUNE. Be aware that FERMI only accepts CILogon certificates.

Online Documentation: [Swagger](#)[Redoc](#)
[REST Tutorial](#)

Vincit qui se vincit.

He conquers who conquers himself.

Database Schema

This product is created over a PostgreSQL database. Two views of the schema are available. One which shows all the tables. There is also a second simpler view of the schema which does not show the security tables and their connections.

[Full Schema](#)
[Simplified Schema](#)
[Database Conceptual Overview](#)

- Should we make it something more obvious for the production?

E.g.,

“You are accessing to the production version...”

Migrating (Copying) from dev. to pro.?

- Would it be helpful if we prepare such script(s) and provide to everybody?
- Or there isn't much demand for this?

Which fields are actually required in Component Type definition?

Hajime Muramatsu U of Minnesota

DUNE HWDB Tutorial JUL/15/2024 42

Completing Component Type

Edit Component Type

SPECS LOG	IMAGES
Type Name	CPA_Parts_FR4_bottom
Type ID	00017
Full Name	Z.Sandbox.Sandbox.CPA_Parts_FR4_bottom
Part Type ID	Z00100100017
Comments	FR4 Frames : Bottom Support Bars
Category	generic
Managed by	x type-manager
Manufacturers	x Hajime Inc
Created	2021-03-25 11:53:35
Created by	Hajime Muramatsu
Specifications	Version 1 Created 2021-03-25 11:53:35-05:00 Created by Hajime Muramatsu
Datasheet	Parts ID: -1 Frame Name: Bottom Support Bar Drawing Number: DFD-20-A601 Number Ordered: -1 Number Received: -1
Connectors	select a type
SAVE	DONE

- In the following several pages, we'll go through;
 - ▶ **"Managed by" (Roles) : Required**
 - ▶ **"Manufacturers" : Optional**
 - ▶ **"Specifications" : Required**
 - ▶ **"Connectors" : Optional**
 - ▶ **"SPECS LOG"**

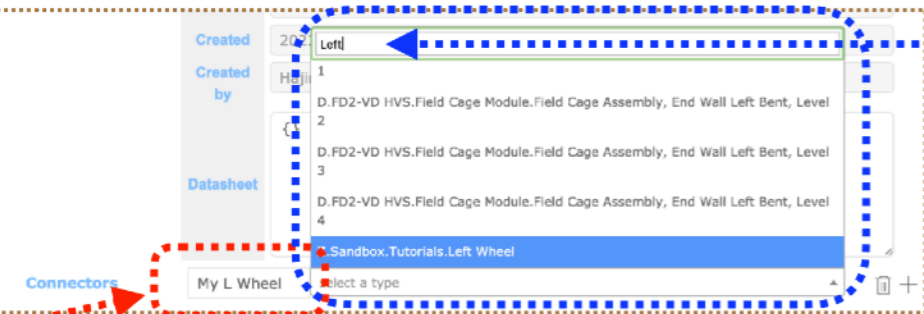
- Updated the slides and the Training site.

Case-sensitive in entered sub-Component Type names

Defining sub-Component Types

WANT Left Wheel and Right Wheel to be sub-components of Front Axle

Provide the corresponding sub-Component Type Name (they are case-sensitive!)
Entering a few characters will give you a list of matched candidates.

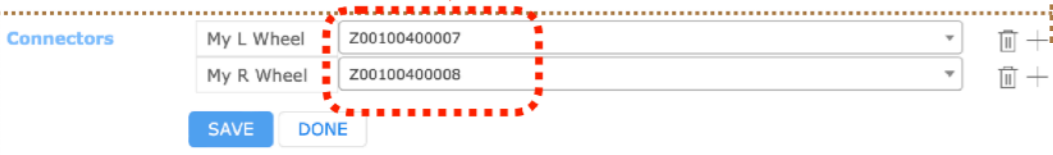


Give your preferred name for its functional position (can be any, but needs to be unique within this Component)



- Those characters entered in defining sub-Component Type names are case-sensitive.

Updated the slides and the Training site.



Saving triggers the conversion from Type Names to Type IDs.

That is it! You are done!

Treatment of sub-Component PIDs in the case of shipment box

- In general, we recommend to assign a PID to your shipment box. This allows you to track its latest location and its history of location.
 - I was wrong to suggest to assign PIDs to its contents as sub-components. Problem in this case is that those contents (sub-components) will be very likely sub-components of another assembled component, instead of just this box!
 - Our suggestion in this case is to store those sub-Component PIDs in “Item Specifications”.
- This way, one can still retrieve those sub-Component PIDs when s/he scans a QR-code of shipment box.
- Scanning such QR-code will take you to the shipment box Item page, which then shows its Item Specifications. See the next page....

- QR
- LOCATION LOG
- SPECS LOG
- STRUCTURE LOG
- CONTAINER LOG
- TEST LOG
- IMAGES

Component Type Z.FD1-HD HVS.HWDBUnitTest.Test Type 002

Part ID Z00599800002-00001-US186

Location

Serial Number

Country of Origin United States

Resp. Institution University of Minnesota Twin Cities

Manufacturer Hajime Inc

Created 2024-07-19 13:23:13

Created by Hajime Muramatsu

Specs Version 0

Comments

Component Status available

Contained in N/A

Specifications

Name	My Special Shipping Crate
My Tools	D00502005200-00031
Drawing Number	DFD-99-9999
LowMed Panel 1	D00502006200-00100
LowMed Panel 2	D00502006200-00101
LowBott Panel 1	D00502006100-00001
LowBott Panel 2	D00502006100-00002

- This way, we can preserve info of the contents (sub-comp. PIDs), while we are actually not using (linking) them.

Sub-components

SAVE DONE

Image formats that are supported

- jpeg, tiff, pdf, bmp, png...

At least, I have tried these and know they are accepted.

- docx, xlsx are NOT accepted... for now.

Would you like them to be?

oid, can it be also obtained with the WEB-UI?

- Yes.

Take this (Z00100200040-00001) as an example.

From the REST API, we know;

Created @ 2020-11-16T10:13:48.704421-06:00

the old = 110

image_id = 12529424-86a6-11eb-a31b-936fd91bb429

file name = DFD-20-A017.pdf

```
"data": [
  {
    "comments": "here is the 1st test of the test result...",
    "created": "2020-11-16T10:13:48.704421-06:00",
    "creator": {
      "id": 12624,
      "name": "Hajime Muramatsu",
      "username": "hajime3"
    },
    "id": 110,
    "images": [
      {
        "id": "12529424-86a6-11eb-a31b-936fd91bb429",
        "name": "DFD-20-A017.pdf"
      }
    ],
    "link": {
      "href": "/cdbdev/api/v1/component-tests/110",
      "rel": "details"
    },
    "methods": [
      {
        "href": "/cdbdev/api/v1/component-tests/110/images",
        "rel": "Images"
      }
    ],
    "test_data": {
      "cleaned": 0
    }
  }
]
```

- Two test results are shown.

- The 1st one has an image, while the 2nd one does not.

- Notice that, independent of whether there is an image(s) associated with its

Test history for [Z00100200040-00001](#)[ADD NEW TEST...](#)

Test_type_name	Created	Creator	Test_data	Test_spec_version	Comments
CPA_Parts_FR4_main QC check	2020-11-16 10:13:48.704421-06:00	Hajime Muramatsu	Visual: 40 Cleaned: 0 Comments: null Template: 1	None	here is the 1st test of the test result...
CPA_Parts_FR4_main QC check	2020-11-16 10:13:18.708351-06:00	Hajime Muramatsu	Visual: 40 Cleaned: 0 Comments: All looked/passed good... Template: 1	None	here is the 1st test of the test result...
CPA_Parts_FR4_main QC check	2020-11-16 10:12:46.909506-06:00	Hajime Muramatsu	Visual: 40 Cleaned: 50 Comments: All looked/passed good... Template: 50	None	here is the 1st test of the test result...
CPA_Parts_FR4_main QC check	2020-10-23 11:02:43.333105-05:00	Hajime Muramatsu	Visual: 50 Cleaned: 50 Comments: All looked/passed good... Template: 50	None	here is the 1st test of the test result...

- Goto the TestLog of the Item.
- Click the corresponding Test entry.

- The click "IMAGES"

Edit Test for [Z00100200040-00001](#)

TEST LOG IMAGES

Component Type: CPA_Parts_FR4_main

Test Type: CPA_Parts_FR4_main QC check

Spec Version: -1

Comments: here is the 1st test of the test result...

Created: 2020-11-16 10:13:48

Created by: Hajime Muramatsu

Test Results

Visual: 40

Cleaned: 0

Comments:

Template: 1

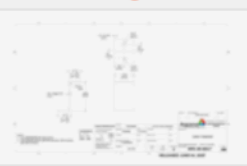
SAVE DONE

Images

< Previous

For test with comment: [here is the 1st test of the test result...](#)

ADD NEW... FILTER...

Image_name	Library	Item	Comments	Created	Image	Creator
DFD-20-A017.pdf	test		"here is the 1st test of the test result..."	2020-11-25 14:30:52UTC-06:00		Hajime Muramatsu

- At this point, your browser should be showing; https://dbweb9.fnal.gov:8443/cdbdev/view/images/component_test/110 where the last path component, 110, is your oid.
 - The file name can be also seen.
 - If the file name is clicked, your browser should be showing; <https://dbweb9.fnal.gov:8443/cdbdev/edit/image/12529424-86a6-11eb-a31b-936fd91bb429>
- Again, the last path component corresponds to its image_id.

Bulk-Add with WEB-UI

- Did you know that when you “bulk-add” items, you receive the corresponding bar/QR-codes like this?
- This is great.
- But would you be interested in obtaining them;
 - in a certain size of pages (e.g., Letter, A4)
 - and while being able to specify specific sizes for each of those labels?

