

PAUL SCHERRER INSTITUT



Stefan Ritt :: Head of Muon Physics Group :: Paul Scherrer Institut

MIDAS recent developments

MIDAS Workshop, Sept. 12, 2023



- OO Access to the ODB: odbxx API
- ODB Editor
- Improvements to sequencer
- Improvements to midas history
- MSetPoint

New Idea

- Inspired by Niels Lohmann: JSON for modern C++
<https://github.com/nlohmann/json>

```
#include <nlohmann/json.hpp>

using json = nlohmann::json;

int main()
{
    json j = {
        { "pi", 3.141 },
        { "happy", true },
        { "answer", {
            { "everything", 42 }
        } },
        { "name", "midas" },
        { "list", {1, 0, 2} }
    };

    j["pi"] = 3.14156;
    std::cout << j["pi"] << std::endl;
}
```

- Just one header file
- Vanilla C++11
- JSON object almost feel like native C++ literal
- Use JSON like an STL container
- Automatic type deduction

Aug 7, 2019

MIDAS workshop, TRIUMF

New C++ ODB API "mdata"

```
#include <mdata.hpp>

using mdata = midas::mdata;

int main()
{
    cm_connect_experiment(...);

    mdata d("/Equipment/Trigger/Common");

    std::cout << d["Format"] << std::endl;

    d["Event ID"] = 1;

    d["Format"].addListener([](mdata &d) {
        std::cout << d << std::endl;
    });
}
```

"d" contains a local copy of ODB subtree

send data to ODB

hot-links via lambda

Aug 7, 2019

MIDAS workshop, TRIUMF

One COVID-19 later...

```
#include <odbxx.h>

int main()
{
    cm_connect_experiment(...);

    midas::odb o = {
        {"Int32 Key", 42},
        {"Bool Key", true}
    }
    o.connect("/Test/Settings");
    o["Int32 Key"] = 43;
    o["Int32 Key"]++;
    o["Bool Key"] = false;

    std::cout << o << std::endl;

    o.watch([](midas::odb &d) {
        std::cout << d << std::endl;
    });
}
```

Create subtree in ODB

Access elements type-safe

JSON Serializer

ODB Hot-link

Thanks to Nik Berger, Uni Mainz

```

// set, retrieve, and change ODB value
o["Int32 Key"] = 42;
int i = o["Int32 Key"];
o["Int32 Key"] = i+1;
o["Int32 Key"]++;
o["Int32 Key"] *= 1.3;
std::cout << "Should be 57: " << o["Int32 Key"] << std::endl;

// test with bool
o["Bool Key"] = false;
o["Bool Key"] = !o["Bool Key"];

// test with std::string
o["Subdir"]["Subsub"]["String Key"] = "Hello";
std::string s = o["Subdir"]["Subsub"]["String Key"];
s += " world!";
o["Subdir"]["Subsub"]["String Key"] = s;
s = s + o["Subdir"]["Subsub"]["String Key"].s(); // need .s() for concatenations

// test with a vector
std::vector<int> v = o["Int Array"]; // read vector
std::fill(v.begin(), v.end(), 10);
o["Int Array"] = v; // assign vector to ODB array
o["Int Array"][1] = 2; // modify array element
i = o["Int Array"][1]; // read from array element
o["Int Array"].resize(5); // resize array
o["Int Array"]++; // increment all values of array
std::cout << "Arrays size is " << o["Int Array"].size() << std::endl;

// auto-enlarge arrays
o["Int Array"][10] = 10;

// test with a string vector
std::vector<std::string> sv;
sv = o["String Array"];
sv[1] = "New String";
o["String Array"] = sv;
o["String Array"][2] = "Another String";
o["String Array"][3] = std::string("One more");
s = o["String Array"][1].s(); // need .s() to explicitly convert to std::string

```

```

// test with strings with given size
o["String Array 2"][0].set_string_size("Hello", 64);
o["String Array 2"][1] = "Second string";
o["String Array 2"][2] = "Third string";

// test with bool arrays/vectors
o["Bool Array"] = std::array<bool, 3>{true, false, true};
o["Bool Array from Vector"] = std::vector<bool>{true, false, true};

// iterate over array
int sum = 0;
for (int e : o["Int Array"])
    sum += e;
std::cout << "Sum should be 37: " << sum << std::endl;

// create key from other key
midas::odb oi(o["Int32 Key"]);
oi = 123;

// test auto refresh read
std::cout << oi << std::endl; // each read access reads value from ODB
oi.set_auto_refresh_read(false); // turn off auto refresh
std::cout << oi << std::endl; // this does not read value from ODB
oi.read(); // this forces a manual read
std::cout << oi << std::endl;

// test auto refresh write
oi.set_auto_refresh_write(false); // turn off auto refresh write
oi = 321; // this does not write a value to the ODB
oi.write(); // this forces a manual write

// create ODB entries on-the-fly
midas::odb ot;
ot.connect("/Test/Settings/OTF");// this forces /Test/OTF to be created if not already
ot["Int32 Key"] = 1; // create all these keys with different types
ot["Double Key"] = 1.23;
ot["String Key"] = "Hello";
ot["Int Array"] = std::array<int, 10>{};
ot["Subdir"]["Int32 Key"] = 42;
ot["String Array"] = std::vector<std::string>{"S1", "S2", "S3"};

```

```

// create key with default value
i = ot["Int32 Key"](123); // key exists already (created above) -> use key value
i = ot["New Int32 Key"](123); // key does not exist -> set it to default value 123
// std::string s1 = ot["New String Key"]("Hi"); // same for strings
std::cout << ot << std::endl;

// re-read the underlying ODB tree which got changed
OTF code
std::cout << o.print() << std::endl;

// iterate over sub-keys
for (midas::odb& oit : o)
    std::cout << oit.get_name() << std::endl;

// print whole sub-tree
std::cout << o.print() << std::endl;

// print whole subtree
std::cout << o.dump() << std::endl;

// update structure - create keys if needed, keep existing values if key already exists
// delete keys that are in ODB but not the list of defaults.
midas::odb o3 = {
    {"Int32 Key", 456},
    {"New Bool Key", true},
    {"String Array", {"Hello1", "Hello2", "Hello3"}},
    {"Bool Key", true},
    {"Subdir", {
        {"Int32 key", 135 },
        {"New Sub Bool Key", false},
        {"Double Key", 1.5}
    }}
};
o3.connect_and_fix_structure("/Test/Settings");

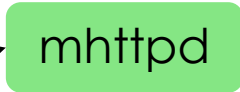
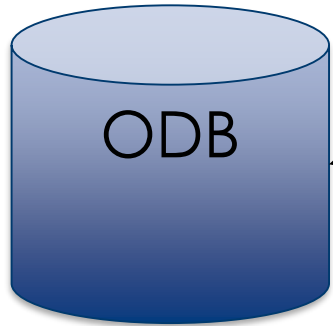
// Print new structure
std::cout << "After changing structure with o3:" << std::endl;
std::cout << o3.print() << std::endl;

// delete test key from ODB
o3.delete_key("Int32 Key");

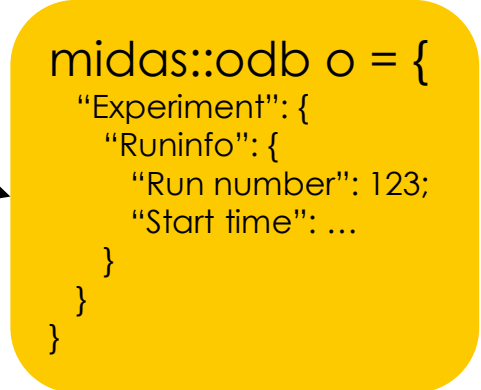
```

[~/midas/examples/odbxx/odbxx_test.cxx](#)

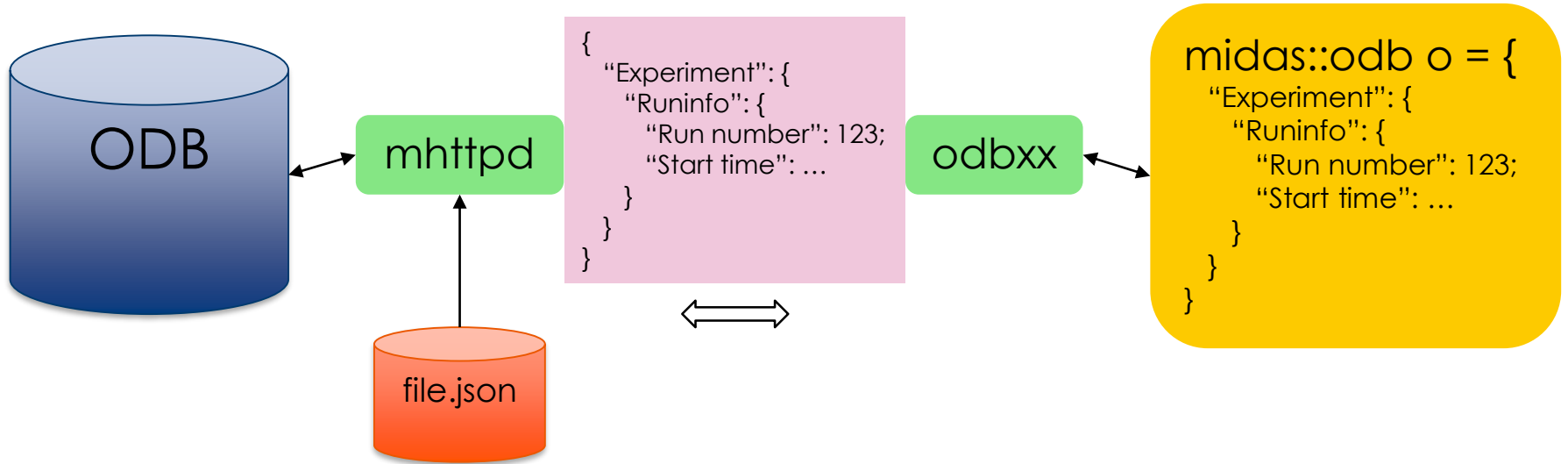
Fetching large ODB trees



"Experiment" ↔
"Runinfo" ↔
"Run number" ↔
"Start Time" ↔



Fetching large ODB trees





Main page
Documentation
Installation
About
Community Support
MediaWiki help

Tools
What links here
Related changes
Upload file
Special pages
Printable version
Permanent link
Page information

Page Discussion

Read Edit View history More

Search MidasWiki

Odbxx

A C++11 object-oriented interface to the ODB ([online database](#)) was introduced in May 2020. You can think of it like a "magic" map/dictionary that automatically sends changes you make to the ODB, and receives updates that others have made.

The header for this odbxx interface is at [odbxx.h](#)[#] and example usage in [odbxx_test.cpp](#)[#].

You can find more details about the ODB on the [ODB Access and Use](#) page, which includes links to the command-line, javascript, python, and non-object C++ interfaces.

Contents [hide]

- 1 Basic usage
- 2 Automatic refreshing
- 3 Arrays/vectors
- 4 Strings
- 5 Creating new bits of the ODB
- 6 Iterating over subkeys
- 7 Callback functions
- 8 Utility functions
 - 8.1 void odb::create(const char *name, int type)
 - 8.2 void odb::delete_key()
 - 8.3 int odb::delete_key(const std::string &name)
 - 8.4 bool odb::exists(const std::string &name)
 - 8.5 void odb::exists(const std::string &name)
 - 8.6 void odb::set_debug(bool flag) / bool odb::get_debug()
- 9 Example code

Basic usage [edit]

The simplest usage is like:

```
// Grab a bit of the ODB
midas:odb exp("/Experiment");

// Simple read
std::cout << "The current transition timeout is " << exp["Transition timeout"] << std::endl;

// Make a change. The new value is automatically sent to the ODB.
// Most C++ operators are supported (++ ,+= etc), or you can do a simple
// re-assignment like "exp["Transition timeout"] = 12345;".
exp["Transition timeout"] += 100;

// Read the new value
std::cout << "The transition timeout is now " << exp["Transition timeout"] << std::endl;
```

You can automatically cast to regular data types (int, double) etc if you want a copy of the value to work with:

```
int curr_timeout = exp["Transition timeout"];
```

Note: The ODB directory you connect to ("/Experiment" in the above example), has to start with a "/" to tell the midas:odb object to connect directly to the ODB. Otherwise, a simple local midas:odb string object gets created without any connection to the ODB.

Automatic refreshing [edit]

You may temporarily disable the automatic updating to/from the ODB using `odb::set_auto_refresh_write(false)` and `odb::set_auto_refresh_read(false)`.

If auto-refresh is enabled (the default), your new values are sent to the ODB as soon as you touch the value in the `midas:odb` object. The ODB is queried for new values whenever you access the value. In the above example, the ODB is queried 4 times (during construction of `exp` and each time `exp["Transition timeout"]` is mentioned), and written to 1 time (when `exp["Transition timeout"]` is assigned to).

See the [#Callback functions](#) section below for details on how to have a function called when a value changes.

Thanks to Ben Smith, TRIUMF

Comparison ODB APIs

old C API

```
cm_get_experiment_database(&hDB, NULL);

size = sizeof(value);
db_get_value(hDB, 0, "/Experiment/Menu/Status",
             &value, &size, TID_BOOL, TRUE);

db_set_value(hDB, 0, "/Experiment/Menu/Status",
             &value, size, 1, TID_BOOL);
```

odbxx API

```
midas::odb o("/Experiment/Menu");

value = o["Status"];

o["Status"] = value;
```

<https://daq00.triumf.ca/MidasWiki/index.php/Odbxx>

MVODB API

```
cm_get_experiment_database(&hDB, NULL);
gOdb = MakeMidasOdb(hODB);

gOdb->Chdir("/Experiment/Menu");
gOdb->RB("Status", &value);
gOdb->WB("Status", value);
```

[https://daq00.triumf.ca/MidasWiki/index.php/Frontend_user_code_\(object_oriented_-_TMFE\)](https://daq00.triumf.ca/MidasWiki/index.php/Frontend_user_code_(object_oriented_-_TMFE))

Conversion of static midas pages

Page	Status
Status	OK
Transition	OK
ODB	OK
Messages	OK
Chat	OK
Elog	OK
Alarms	OK
Programs	OK
Buffers	New
Event Dump	New
History	OK (editor soon)
Equipment	TBD
Sequencer	OK
Config	OK
Help	OK

The screenshot shows the MEG interface with the following components:

- Status Bar:** Alarms: None, 12 Sept 2023, 11:46:36 UTC+2
- Event Selection:**
 - Buttons: Next event, Run, Show only recent events (checked)
 - Fields: Buffer: SYSTEM, Event ID: :1:ALL, Trigger mask: :1
- Event #2895:**
 - Event ID: 1, Trigger Mask: 0 / 0x0, Serial Number: 2895, Size [Bytes]: 5441272 / 0x5306f8, Time stamp: 12/09/2023, 08:07:39
 - Bank: DRSV, Size [Bytes]: 5319514 / 0x512b5a, Type: UINT8
 - Hex dump of event data (hex and ASCII columns).
- Event Dump:**
 - Bank: DRST, Size [Bytes]: 112560 / 0x1b7b0, Type: UINT8
 - Hex dump of event data.
- Equipment Information:**
 - Equipment: HV XEC PMT, HV XEC MPPC, HV TC, HV Cosmic, HV BGO, DC HV, BPS, Xenon, LED, TC, Laser, LHC, COBRA, NAI, RDC, ...
 - Groups: All, Input, ODB, Common, Settings, Variables
 - Table with columns: Names, Input

Equipment: HV XEC PMT, HV XEC MPPC, HV TC, HV Cosmic, HV BGO, DC HV, BPS, Xenon, LED, TC, Laser, LHC, COBRA, NAI, RDC, ...

Groups: All, Input, ODB, Common, Settings, Variables

Names	Input
Air conditioner temp.	18.70322
Rack 0 temp.	30.81418
Rack 1 temp.	-90.99835
Rack 2 temp.	33.64102
Rack 3 temp.	33.75195

Computer Name: NaN



New ODB Editor

The image displays the 'New ODB Editor' interface with several windows and their interactions:

- Create ODB entry:** A dialog box for creating a new entry. Fields include Directory: /Experiment, Type: String, Name: Test, Array size: 1, and String length: 32. Buttons: Create, Cancel.
- Online Database Browser:** The main window showing a tree view of the database structure under 'Experiment'. It includes a toolbar with icons for navigation and editing. A table of key-value pairs is displayed below.
- Results of search for substring "Test":** A search results window showing a list of keys containing the substring 'Test' and their corresponding values.
- ODB open records under "/>**

Thanks to Zaher Salman, PSI

Sequencer improvements

SCRIPT <i>script</i> [, a, b, c, ...]	To call a script on the server side. Optionally, pass parameters to the script.
SET <i>name, value</i> or <i>name = value</i>	Sets the variable <i>name</i> to "value". The variable can then be referenced later in the script by putting a "\$" in front of the name like "\$name". <i>value</i> can be a simple number or a complex expression containing other variables (preceded by a \$) and calculations such as $v + 3 * x * \sin(\$t)$. Following functions are available: <i>abs, acos, asin, atan, atan2, ceil, cos, cosh, e, exp, fac, floor, ln, log, pi, pow, sin, sinh, sqrt, tan, tanh</i> . A variable can be an array by using square brackets, like "a[10] = 5" or "x = \$a[10]".
SUBROUTINE <i>name</i> ...	Declares a subroutine which can be called via CALL.

- <https://daq00.triumf.ca/MidasWiki/index.php/Sequencer>
- See first talk tomorrow on "Mu3e" by Luigi Vigani
- New possibility to run several sequencers in parallel (→ talk Konstantin Olchanski)

```

# Calculate step size
num_steps = $num_steps + 1 # add one step for stop_voltage
step_size = ($stop_voltage-$start_voltage) / ($num_steps-1)

# Initialize measurement arrays at startup
ODBCREATE /Equipment/Test/Variables/Voltage, FLOAT, $num_steps
ODBCREATE /Equipment/Test/Variables/Current, FLOAT, $num_steps
ODBCREATE /Equipment/Test/Variables/V, FLOAT

# Erase any previously stored array
ODDBSET /Equipment/Test/Variables/Voltage[*], 0
ODDBSET /Equipment/Test/Variables/Current[*], 0

v = $start_voltage
ODDBSET /Equipment/Test/Variables/V, $v

current = 0

# Turn on Keithley
ODDBSET /Equipment/KEITHLEY/Variables/Set State, 1
# Wait to turn on
WAIT SECONDS, 2

# Looping starts at 1
LOOP i, $num_steps
  # Store voltage in array and in variable
  ODDBSET /Equipment/Test/Variables/Voltage[$i-1], $v
  ODDBSET /Equipment/Test/Variables/V, $v

  # Set voltage and measure
  ODDBSET /Equipment/KEITHLEY/Variables/Demand Voltage, $v
  # Wait for measurement to be stored in Current
  WAIT SECONDS, 10
  ODDBGET /Equipment/KEITHLEY/Variables/Current, current

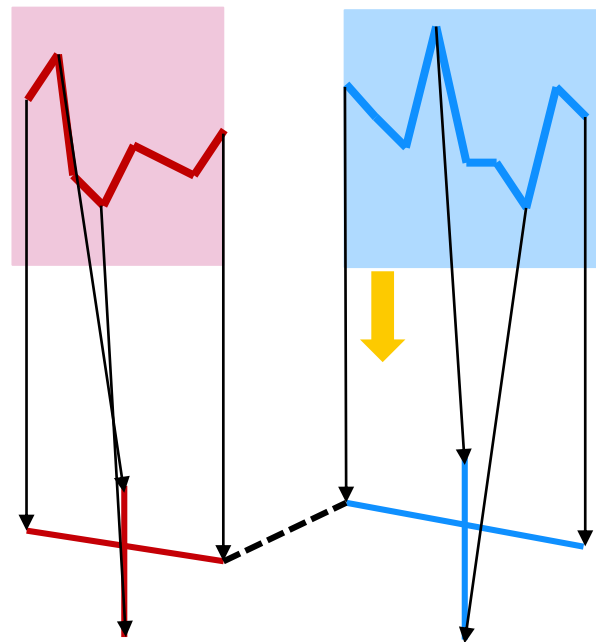
  # Outputting current to ODB array
  ODDBSET /Equipment/Test/Variables/Current[$i-1], $current

  # increment voltage
  v = $v + $step_size
ENDLOOP

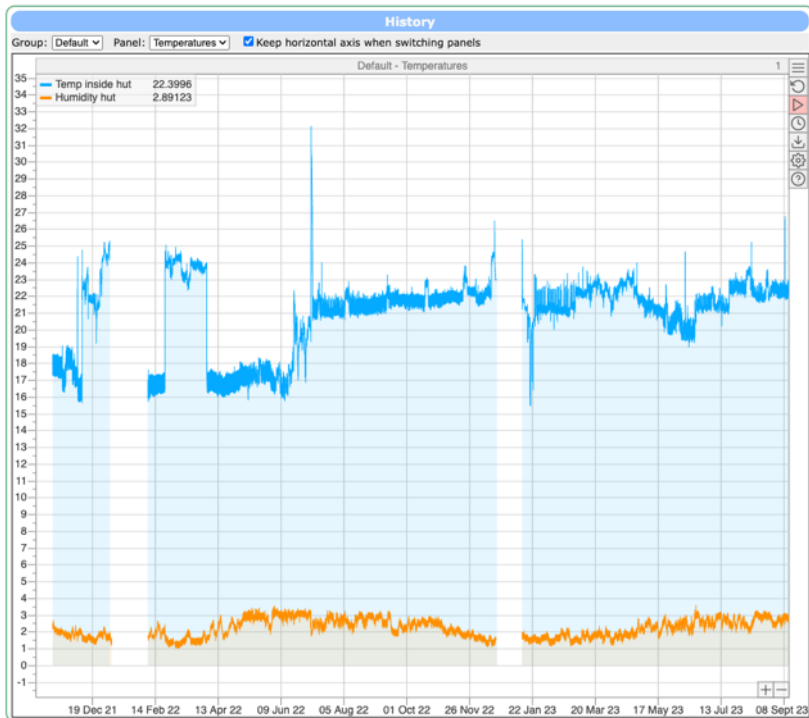
```

Improvements to MIDAS history

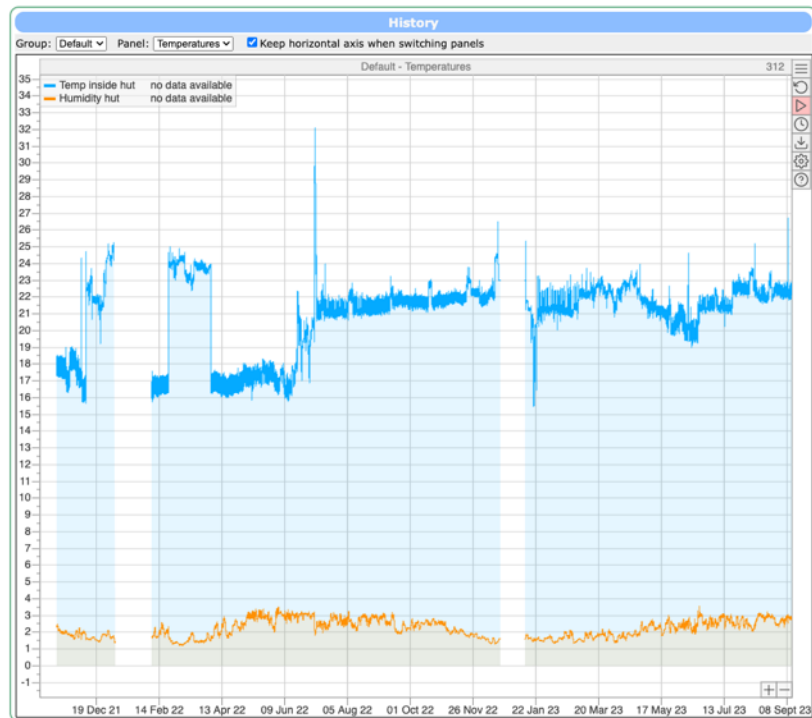
- Plotting **large** history **datasets** can take long and **hit** the **memory limit** of the browser
- **Down-sampling** of history data on server side using first-min-max-last method has been implemented recently
- Web part will be finished in the next days
- A typical history plot of 1y data takes only 1-2 seconds
- See also discussion on “SQL history development” by Gennaro Tortone later today



History comparison



un-binned ~1M data points



binned 10000 "bucket" data points

- **Current** QT-based **SetPoint** solution at PSI approaches **end-of-life**
- **Replaced** by MIDAS **MSetPoint** system:
 - Dedicated “**general**” EPICS front-end
 - EPICS **CA definition** in ODB
 - **Custom page** to control beamline elements

The screenshot shows the 'Setpoint V2.7.3-0-gaf466c9' application window. The title bar indicates the file path: /hipasec/data/AppData/pie5/MEG2022-COBRA-final-150kV.sp. The window contains a table with columns: Device, Description, Inv, File value, Set value, Current value, Stc, StatusX, Command, and Scale. The table lists various beamline elements such as Beamline Pie5, MHC4-IST:2, ZPESCNTINP2, ZPESCNTINP3, AHSW41-IST:2, QSF41, HSC41, QSF42, QSF43, HSC42, QSF44, QSF45, HSC43, QSF46, QSF47, HSC44, QSF48, AST41, HSC41, SSK41, OSB41, OSB42, OSB43, SEP41, QSK41, QSK42, QSK43, SML41, SMH41, SMV41, FSH41-L, FSH41-R, FS41-L, FS42-R, FS42-V, FSH43-L, FSH43-R, and KSF41. The 'Current value' column shows values like 0.00, 93.26, 2.00, 83.99, -65.00, 0.00, 81.99, 85.99, 36.00, -81.54, 94.59, -25.00, -51.48, -97.40, 113.76, 0.00, 37.76, -48.00, 15.00, -72.00, 18.50, -44.10, 43.00, -25.00, 200.00, 72.50, 30.00, 30.00, 80.00, 80.00, 160.00, 120.00, 120.00, 150.00, 150.00, 150.00, 1.000, and 1.000. The 'StatusX' column shows various status indicators like 'A', 'equ', and 'PSA Not OK'. The 'Command' column shows 'SET ZERO' and 'Magnet ON/OFF'. The bottom status bar indicates: Success: loaded /hipasec/data/AppData/pie5/MEG2022-COBRA-final-150kV.sp (updates/Items:25 Changes:25) 203Pv(68NC) / 2023-Feb-09 09:48:59.

Device	Description	Inv	File value	Set value	Current value	Stc	StatusX	Command	Scale
1	Beamline Pie5								
2	MHC4-IST:2				0.00	µA			
3	ZPESCNTINP2	Gate 100Hz/A			0.00	Hz			
4	ZPESCNTINP3	Scaler			0.00	Hz			
5	AHSW41-IST:2				0.00	A			
6	QSF41		93.26	0.00	0.00	A			
7	HSC41		2.00	0.00	0.00	A			
8	QSF42		83.99	0.00	0.00	A			
9	QSF43		-65.00	0.00	0.00	A			
10	HSC42		0.00	0.00	0.00	A			
11	QSF44		81.99	0.00	0.00	A			
12	QSF45		85.99	0.00	0.00	A			
13	HSC43		36.00	0.00	0.00	A			
14	QSF46		-81.54	0.00	0.00	A			
15	QSF47		94.59	0.00	0.00	A			
16	HSC44		-25.00	0.00	0.00	A			
17	QSF48		-51.48	0.00	0.00	A			
18	AST41		-97.40	0.00	0.00	A			
19	HSC41		113.76	0.00	0.00	A			
20	SKK41		0.00	0.00	0.00	A			
21	OSB41		37.76	0.00	0.00	A			
22	OSB42		-48.00	0.00	0.00	A			
23	OSB43		15.00	0.00	0.00	A			
24	SEP41		-72.00	0.00	0.00	A			
25	QSK41		18.50	0.00	0.00	A			
26	QSK42		-44.10	0.00	0.00	A			
27	QSK43		43.00	0.00	0.00	A			
28	SML41		-25.00	0.00	0.00	A			
29	SMH41		200.00	0.00	0.00	A			
30	SMV41		72.50	0.00	0.00	A			
31	FSH41-L		30.00	0.00	0.00	equ	<<NC>>	equ	
32	FSH41-R		30.00	0.00	0.00	equ	<<NC>>	equ	
33	FS41-L		80.00	0.00	0.00	equ	<<NC>>	equ	
34	FS42-R		80.00	0.00	0.00	equ	<<NC>>	equ	
35	FS42-V		160.00	0.00	0.00	equ	<<NC>>	equ	
36	FSH43-L		120.00	0.00	0.00	equ	<<NC>>	equ	
37	FSH43-R		120.00	0.00	0.00	equ	<<NC>>	equ	
38									
39	SEP41 HV block (be patient...)								
40	SEP41VHVP-STA:2	HV+ Status OK?					<<NC>>	equ	
41	SEP41VHVP	HV+	150.00	0.00	0.00	equ	<<NC>>	equ	
42	SEP41VHVPJSTI:2	Actual I+					<<NC>>	equ	
43	SEP41VHVN-STA:2	HV+ Status OK?					<<NC>>	equ	
44	SEP41VHVN	HV+	150.00	0.00	0.00	equ	<<NC>>	equ	
45	SEP41VHVNJSTI:2	Actual I+					<<NC>>	equ	
46	SEP41VHV	Don't forget to set HV							
47									
48	KSF41:STA:1						PSA Not OK		
49	KSF41								
50	KSV41:STA:1						PSA Not OK		
51	KSV41								

old QT application

Save / Load current settings on server

Export / Import current settings on client

Apply file values with scaling

Currently loaded file

Send command to selected device(s)

Selected element for command

Custom pages with sub-menus

MEG Alarms: None 12 Sept 2023, 15:04:35 UTC+2

Command Save Load Export Import ? msetpoint/MEGII_CC_final_okv_allelementsToZero_SlitsFSH41_5.json

Device	Description	Status	File value	Set value	Measured
0	AHSW41	Value	On		97.14 A
1	QSF4	Magnet	On	0.00 = 0.00	-0.06 A
2	HSC41	Magnet	On	0.00 = 0.00	-0.04 A
3	QSF42	Magnet	On	0.00 = 0.00	0.30 A
4	QSF43	Magnet	On	0.00 = 0.00	-0.03 A
5	HSC42	Magnet	On	0.00 = 0.00	-0.00 A
6	QSF44	Magnet	On	0.00 = 0.00	-0.07 A
7	QSF45	Magnet	On	0.00 = 0.00	-0.04 A
8	HSC43	Magnet	On	0.00 = 0.00	-0.06 A
9	QSF46	Magnet	On	0.00 = 0.00	-0.10 A
10	QSF47	Magnet	On	0.00 = 0.00	-0.09 A
11	HSC44	Magnet	On	0.00 = 0.00	-0.01 A
12	QSF48	Magnet	On	0.00 = 0.00	-0.08 A
13	AST41	Magnet	On	0.00 = 0.00	-1.61 A
14	ASC41	Magnet	On	0.00 = 0.00	1.55 A
15	SSL41	Magnet	On	0.00 = 0.00	0.00 A
16	QSB41	Magnet	On	0.00 = 0.00	1.62 A
17	QSB42	Magnet	On	0.00 = 0.00	-1.91 A
18	QSB43	Magnet	On	0.00 = 0.00	-0.00 A
19	SEP41	Magnet	On	0.00 = 0.00	-1.63 A
20	QSK41	Magnet	On	0.00 = 0.00	0.01 A
21	QSK42	Magnet	On	0.00 = 0.00	0.01 A
22	QSK43	Magnet	On	0.00 = 0.00	0.01 A
23	SML41	Magnet	On	0.00 = 0.00	0.00 A
24	SMH41	Magnet	On	0.00 = 0.00	-0.00 A
25	SMV41	Magnet	On	0.00 = 0.00	-0.00 A
26	FSH41-L	Slit	Remote	5.0 = 5.0	5.0 mm
27	FSH41-R	Slit	Remote	5.0 = 5.0	5.0 mm
28	FS42-L	Slit	Remote	80.0 = 80.0	80.1 mm
29	FS42-R	Slit	Remote	80.0 = 80.0	80.1 mm
30	FS42-V	Slit	Remote	160.0 = 160.0	160.1 mm
31	FSH43-L	Slit	Remote	120.0 = 120.0	119.9 mm

Select command

On

Off

Reset

Reset-DSP

HV On

HV Off

HV Set

Cancel

Select file dialog

msetpoint/

Name	Modified
MEGII_CC_final_okv_allelementsToZero_SlitsFSH41_5.json	Tue Sep 12 2023 09:23:29
meg.json	Thu Aug 17 2023 13:44:59
MEGII_CC_final_150kV_2mA.json	Thu Aug 17 2023 13:44:54
MEGII_CC_final_150kV_1.8mA_20230812.json	Thu Aug 17 2023 13:44:44

Load Cancel



Title for web page

Loaded config file

Increment / Decrement values

Save / Load current config on server

Export / Import current config on client

Add, delete, re-order elements

Title: MSetPoint PIE5 Filename: pie5.json EPICS Gateway: hipa-cagw:5062 mpc2037:5064

Save Load Export Import

Config	Name	Description	Group	Type	Unit	Format	+/-	Warning thr.	PV name	Set	Get	Status	Command	History
0	AHSW41	(empty)		Value	A	(empty)	0	0.5	AHSW41	(empty)	:IST:2	:STA:2	(empty)	(empty)
1	QSF41	(empty)		Magnet	A	(empty)	1	0.5	QSF41	:SOL:2	:IST:2	:STA:2	:COM:2	(empty)
2	HSC41	(empty)		Magnet	A	(empty)	1	0.5	HSC41	:SOL:2	:IST:2	:STA:2	:COM:2	(empty)
3	QSF42	(empty)		Magnet	A	(empty)	1	0.5	QSF42	:SOL:2	:IST:2	:STA:2	:COM:2	(empty)
4	QSF43	(empty)		Magnet	A	(empty)	1	0.5	QSF43	:SOL:2	:IST:2	:STA:2	:COM:2	(empty)
5	HSC42	(empty)		Magnet	A	(empty)	1	0.5	HSC42	:SOL:2	:IST:2	:STA:2	:COM:2	(empty)
6	QSF44	(empty)		Magnet	A	(empty)	1	0.5	QSF44	:SOL:2	:IST:2	:STA:2	:COM:2	(empty)
7	QSF45	(empty)		Magnet	A	(empty)	1	0.5	QSF45	:SOL:2	:IST:2	:STA:2	:COM:2	(empty)
8	HSC43	(empty)		Magnet	A	(empty)	1	0.5	HSC43	:SOL:2	:IST:2	:STA:2	:COM:2	(empty)
9	QSF46	(empty)		Magnet	A	(empty)	1	0.5	QSF46	:SOL:2	:IST:2	:STA:2	:COM:2	(empty)
10	FSH41-L	(empty)		Slit	mm	(empty)	10	0.5	FSH41-L	:SOL	:POSA	:STA:2	:COM:2	(empty)
11	FSH41-R	(empty)		Slit	mm	(empty)	10	0.5	FSH41-R	:SOL	:POSA	:STA:2	:COM:2	(empty)
12	FS42-L	(empty)		Slit	mm	(empty)	10	0.5	FS42-L	:SOL	:POSA	:STA:2	:COM:2	(empty)
13	FS42-R	(empty)		Slit	mm	(empty)	10	0.5	FS42-R	:SOL	:POSA	:STA:2	:COM:2	(empty)
14	Beam Blocker (1=open)	(empty)		Beamblocker	(empty)	(empty)	0	0.5	KSF41	:COM:2	:STA:1	:STA:2	:COM:2	EPICS/Proton Current
15	PSA (1=ready)	(empty)		PSA	(empty)	(empty)	0	0.5	KSF41	(empty)	:STA:1	:STA:2	:COM:2	(empty)
16	Proton current	(empty)		Value	uA	(empty)	0	0.5	MHC4	(empty)	:IST:2	(empty)	(empty)	EPICS/Proton Current
17	Sep. HV- voltage	(empty)		Separator	kV	(empty)	10	0.5	SEP41VHVN	:SOLV:2	:ISTV:2	:STA:2	:COM:2	(empty)
18	Sep. HV- current	(empty)		Value	mA	(empty)	0	0.5	SEP41VHVN	:SOLI:2	:ISTI:2	:STA:2	(empty)	(empty)
19	Sep. HV+ voltage	(empty)		Separator	kV	(empty)	10	0.5	SEP41VHVP	:SOLV:2	:ISTV:2	:STA:2	:COM:2	(empty)
20	Sep. HV+ current	(empty)		Value	mA	(empty)	0	0.5	SEP41VHVP	:SOLI:2	:ISTI:2	:STA:2	(empty)	(empty)
21	Sep vacuum	(empty)		Value	mbar	(empty)	0	0.5	SEP41VHV	(empty)	:VAC2:2	(empty)	(empty)	(empty)
22	Sep. Xray counter	(empty)		Value	Hz	(empty)	0	0.5	SEP41VHV	(empty)	:XRAY:2	(empty)	(empty)	(empty)
23	Beamline Power	(empty)		Value	kW	(empty)	0	0.5	ZPIE5	(empty)	:TOTPWR:2	(empty)	(empty)	(empty)

MIDAS mode vs. Standalone mode in MEG

MEG

Alarms: None 17 Aug 2023, 09:52:14 UTC+2

msetpoint/MEGIL_CC_fmal_150kV_1.8mA_20230812.json

Device	Description	Status	File value	Set value	Measured		
0	AHSW41	Value	On	97.14	A		
1	QSF41	Magnet	On	-93.25	-93.25	-93.29	A
2	HSC41	Magnet	On	-12.00	-12.00	-11.87	A
3	QSF42	Magnet	On	82.62	82.62	82.63	A
4	QSF43	Magnet	On	-64.45	-64.45	-64.47	A
5	HSC42	Magnet	On	-26.00	-26.00	-26.01	A
6	QSF44	Magnet	On	81.81	81.81	81.82	A
7	QSF45	Magnet	On	85.00	85.00	84.94	A
8	HSC43	Magnet	On	52.00	52.00	52.04	A
9	QSF46	Magnet	On	-82.66	-82.66	-82.69	A
10	QSF47	Magnet	On	95.15	95.15	95.16	A
11	HSC44	Magnet	On	-29.00	-29.00	-28.97	A
12	QSF48	Magnet	On	-50.57	-50.57	-50.54	A
13	AST41	Magnet	On	-97.40	-97.40	-97.41	A
14	ASC41	Magnet	On	113.78	113.78	113.84	A
15	SSL41	Magnet	On	-1.00	-1.00	-1.00	A
16	QSB41	Magnet	On	37.80	37.80	37.84	A
17	QSB42	Magnet	On	-48.00	-48.00	-47.98	A
18	QSB43	Magnet	On	15.00	15.00	15.00	A
19	SEP41	Magnet	On	-73.00	-73.00	-72.94	A
20	QSK41	Magnet	On	25.50	25.50	25.50	A
21	QSK42	Magnet	On	-47.50	-47.50	-47.51	A
22	QSK43	Magnet	On	43.20	43.20	43.21	A
23	SM41	Magnet	On	-15.00	-15.00	-14.99	A
24	SMH41	Magnet	On	200.00	200.00	200.00	A
25	SMV41	Magnet	On	190.00	190.00	190.00	A
26	FSH41-L	Slit	Remote	22.5	22.5	22.4	mm
27	FSH41-R	Slit	Remote	22.5	22.5	22.6	mm
28	FS42-L	Slit	Remote	80.0	80.0	80.1	mm
29	FS42-R	Slit	Remote	80.0	80.0	80.1	mm
30	FS42-V	Slit	Remote	160.0	160.0	160.1	mm
31	FSH43-L	Slit	Remote	120.0	120.0	120.0	mm
32	FSH43-R	Slit	Remote	120.0	120.0	119.9	mm
33	Beam Blocker	1 = open	Open			1	
34	PSA	1 = ready	Open			1	
35	Proton current	Value				1727.6	uA
36	Sep. HV- voltage	Separator	True	150.00	150.00	150.00	kV
37	Sep. HV- current	Value	True			0.40	mA
38	Sep. HV+ voltage	Separator	True	150.00	150.00	150.00	kV
39	Sep. HV+ current	Value	True			0.00	mA
40	Sep vacuum	Value				2.42e-6	mbar
41	Sep. Xray counter	Value				5.7	Hz
42	Beamline Power	Value				55.4	kW

oddbit -c "load msetpoint.json"



MSetPoint PIE5

Alarms: None 15 Aug 2023, 11:41:04 UTC+2

meg.json

Device	Description	Status	File value	Set value	Measured		
0	AHSW41	Value	On	97.13	A		
1	QSF41	Magnet	On	-93.25	-93.25	-93.22	A
2	HSC41	Magnet	On	-12.00	-12.00	-12.04	A
3	QSF42	Magnet	On	82.62	82.62	82.66	A
4	QSF43	Magnet	On	-64.45	-64.45	-64.44	A
5	HSC42	Magnet	On	-26.00	-26.00	-25.99	A
6	QSF44	Magnet	On	81.81	81.81	81.83	A
7	QSF45	Magnet	On	85.00	85.00	84.97	A
8	HSC43	Magnet	On	52.00	52.00	51.98	A
9	QSF46	Magnet	On	-82.66	-82.66	-82.64	A
10	QSF47	Magnet	On	95.15	95.15	95.20	A
11	HSC44	Magnet	On	-29.00	-29.00	-29.00	A
12	QSF48	Magnet	On	-50.57	-50.57	-50.56	A
13	AST41	Magnet	On	-97.40	-97.40	-97.41	A
14	ASC41	Magnet	On	113.78	113.78	113.81	A
15	SSL41	Magnet	On	-1.00	-1.00	-1.00	A
16	QSB41	Magnet	On	37.76	37.76	37.81	A
17	QSB42	Magnet	On	-48.00	-48.00	-47.98	A
18	QSB43	Magnet	On	15.00	15.00	15.00	A
19	SEP41	Magnet	On	-73.00	-73.00	-73.00	A
20	QSK41	Magnet	On	25.50	25.50	25.50	A
21	QSK42	Magnet	On	-47.50	-47.50	-47.50	A
22	QSK43	Magnet	On	43.20	43.20	43.20	A
23	SM41	Magnet	On	-15.00	-15.00	-15.00	A
24	SMH41	Magnet	On	200.00	200.00	199.99	A
25	SMV41	Magnet	On	190.00	190.00	190.01	A
26	FSH41-L	Slit	Remote	20.5	22.5	22.5	mm
27	FSH41-R	Slit	Remote	20.5	22.5	22.5	mm
28	FS42-L	Slit	Remote	80.0	80.0	80.0	mm
29	FS42-R	Slit	Remote	80.0	80.0	79.9	mm
30	FS42-V	Slit	Remote	160.0	160.0	160.2	mm
31	FSH43-L	Slit	Remote	120.0	120.0	119.8	mm
32	FSH43-R	Slit	Remote	120.0	120.0	119.9	mm
33	Beam Blocker (1=open)	Beamblocker	Open			1	
34	PSA (1=ready)	PSA	Open			1	
35	Proton current	Value				1731.5	uA
36	Sep. HV- voltage	Separator	True	150.00	150.00	150.00	kV
37	Sep. HV- current	Value	True			0.52	mA
38	Sep. HV+ voltage	Separator	True	150.00	150.00	150.00	kV
39	Sep. HV+ current	Value	True			0.00	mA
40	Sep vacuum	Value				2.38e-6	mbar
41	Sep. Xray counter	Value				5.6	Hz
42	Beamline Power	Value				55.5	kW

epics_fe.cxx

```

for (int i = 0; i < beamline.length; i++) {

    if (demand != std::string("")) {
        std::string name = beamline.settings["CA Name"][i];
        name += beamline.settings["CA Demand"][i];
        status = ca_create_channel(name.c_str(), nullptr,
            nullptr, 0, &(beamline.demand[i]));
    }
}

// install callback for demand changes
beamline.variables["Demand"].watch(demandCallback);

...

demandCallback(midas::odb &o) {
    std::vector<float> newDemand = o;

    for (int i = 0; i < beamline.length; i++) {
        if (newDemand[i] != beamline.demandCache[i]) {
            epics_set_demand(i, newDemand[i]);
            beamline.demandCache[i] = newDemand[i];
        }
    }
}

```

MIDAS ODB

CA Name	*
[0]	AHSW41
[1]	QSF41
[2]	HSC41
[3]	QSF42
[4]	QSF43
[5]	HSC42

CA Demand	*
[0]	(empty)
[1]	:SOL:2
[2]	:SOL:2
[3]	:SOL:2
[4]	:SOL:2
[5]	:SOL:2

Demand	*
[0]	NaN
[1]	-93.250999
[2]	-12
[3]	82.621101
[4]	-64.452599
[5]	-26

MSPConfig

g thr.	PV name	Set	Get	Status
0.5	AHSW41	(empty)	:IST:2	:STA:2
0.5	QSF41	:SOL:2	:IST:2	:STA:2
0.5	HSC41	:SOL:2	:IST:2	:STA:2
0.5	QSF42	:SOL:2	:IST:2	:STA:2
0.5	QSF43	:SOL:2	:IST:2	:STA:2
0.5	HSC42	:SOL:2	:IST:2	:STA:2
0.5	QSF44	:SOL:2	:IST:2	:STA:2
0.5	QSF45	:SOL:2	:IST:2	:STA:2
0.5	HSC43	:SOL:2	:IST:2	:STA:2
0.5	QSF46	:SOL:2	:IST:2	:STA:2
0.5	FSH41-L	:SOL	:POSA	:STA:2
0.5	FSH41-R	:SOL	:POSA	:STA:2
0.5	FS42-L	:SOL	:POSA	:STA:2

- Code at ~/midas/msetpoint
- Documentation at <https://bitbucket.org/tmidas/midas/wiki/MSetPoint>
- Template for other control programs (e.g. high voltage)

ChatGPT knows MIDAS (status Sept 2021)



How do I save the contents of the ODB in the midas daq system?



In the MIDAS Data Acquisition System (DAQ), the Online Database (ODB) is a crucial component that stores configuration settings, state information, and other runtime data. The ODB is organized hierarchically, similar to a file system or a registry, and it can be accessed and modified during runtime.

To save the contents of the ODB:

1. Using the Web Interface:

- If you have the MIDAS web interface running, you can navigate to the ODB page.
- There should be an option to save the ODB to a file. Follow the prompts to save the contents.

2. Using the Command Line:

- You can use the `odbedit` tool, which is a command-line interface to the ODB.
- To save the entire ODB to a file, you can use the `save` command within `odbedit`.

```
bash
$ odbedit
[local:Default:S]/> save <filename>
```

Replace `<filename>` with the desired name for your backup file.

3. Automating the Process:

- If you want to automate the saving process, you can script the `odbedit` commands or use the MIDAS API in your own program to interact with the ODB and save its contents periodically or based on certain triggers.

Remember to regularly back up the ODB, especially before making significant changes to your experiment's configuration or the DAQ system itself. This ensures that you can restore the system to a known state if needed.



Can you write me some c++ code to send a string to the ODB



Certainly! To interact with the MIDAS Online Database (ODB) using C++, you'd typically use the MIDAS API. Here's a basic example of how you might send a string to the ODB using C++:

```
cpp
#include <iostream>
#include "midas.h"

int main() {
    // Initialize MIDAS
    if (cm_connect_experiment("", "", "YourProgramName", NULL) != CM_SUCCESS)
        std::cerr << "Cannot connect to the MIDAS experiment." << std::endl;
    return 1;
}

// Get a handle to the ODB
HANDLE hDB;
cm_get_experiment_database(&hDB, NULL);

// Path in the ODB where you want to set the string
const char* odbPath = "/Your/Path/Here";

// The string you want to send
const char* value = "YourStringValue";

// Set the string in the ODB
int size = strlen(value) + 1; // +1 for the null terminator
if (db_set_value(hDB, 0, odbPath, value, size, 1, YID_STRING) != DB_SUCCESS)
    std::cerr << "Failed to set the ODB value." << std::endl;
cm_disconnect_experiment();
return 1;
}

std::cout << "Successfully set the ODB value." << std::endl;

// Disconnect from the experiment
cm_disconnect_experiment();

return 0;
}
```

Make sure to:

1. Replace `"YourProgramName"` with a suitable name for your program.
2. Replace `"/Your/Path/Here"` with the appropriate path in the ODB where you want to set the string.

Regenerate