USER MANUAL
Slow Control System

Lars Fischer* and Mengqing Wu*

*Deutsches Elektronen-Synchrotron DESY, Notkestr. 85, 22607 Hamburg, Germany

June 25, 2018

This document serves as a how-to manual for the common environmental Slow Control System at DESY test beam facility. This system is has a rack-based hardware, connecting to the test beam common DAQ, EUDAQ2, via a SQL DataBase, MySQL. Its hardware and software will be introduced briefly, with a focus on the following concepts: a step-by-step instruction of the installation and the configuration of the software; a easy-to-follow guide for users who wish to integrate their own slow control inputs to our system, or to the common EUDAQ2 datastream.

Contents

1 Introduction

1.1 Hardware and software

The Slow Control System is currently able to collect connected sensor data in a configurable frequence (i.e. how frequent to update the DataBase variables from sensor data), transfer to the test beam common DAQ, EUDAQ2, and finally write to user data taking events as a standard EUDAQ raw event tag, i.e. not affecting any user customized event definition. For further reading, a detailed description of the system design and a manual for developers, please see the following documents:

- Deliverable Report: https://cds.cern.ch/record/2290758/files/AIDA-2020-D15_3.3.pdf;
- Main User Manual: https://cds.cern.ch/record/2284369/files/AIDA-2020-NOTE-2017-007.pdf.

This document will guide you step by step through the software design and all necessary installations.

Rack and sensors

The DAQ system hardware is a rack-based system which is movable and connected via an Ethernet cable, as shown in Figure 1; it is currently equipped with **10 NTC sensors** for temperature measurements in an operation range of $-20\,°\mathrm{C}$ to 100 celsius [1], and **one DIGI sensor** for complimentary measurements of air pressure, dew point and humidity.

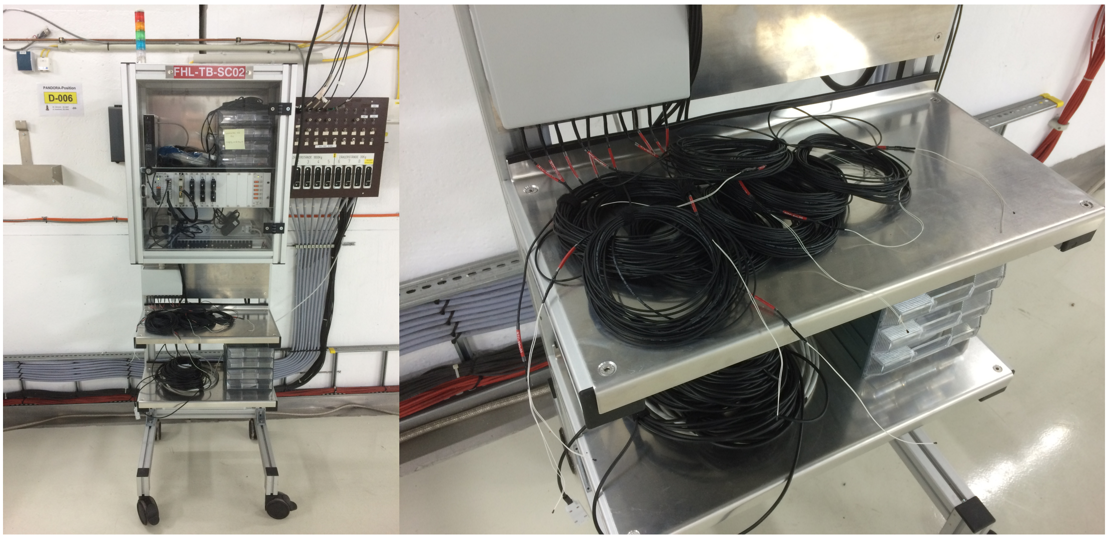


Figure 1: Slow Control System rack #02 (left) in DESY test beam area TB24 and the 10 NTC sensors of 10 m long (right).

The NTC sensors are 10 m long that can be located at different measuring points in a beam area, according to user needs; the DIGI sensor is fixed at the rear of the rack, and can be used as alarm monitor [2]. All the sensors are connected to a data logger (ALMEMO) via a special connector. Users are also able to **mount their own sensors** to the data

[1] NB: to be confirmed, not sure where Lars found this info...

[2] There is **a LED-based alarm** available on the rack (seated on the top of the rack), however, it is not yet plugged to test; users are also able to use it with their defined warning signals.

collector, with many spared connectors stored on the rack, but **please contact the local support team before any customized modification to the system!**

Software

The DAQ software consists of **a commercial DAQ (AMR)** and a testbeam common DAQ software, **EUDAQ2**. The data logger is connected to a Windows-PC mounted on the rack, and it is polled via a USB connector by a commercial DAQ software (AMR) provided by the same company.

This document will give a detailed instruction to guide you to configure and to use the DAQ system to get a measurement based on the connected sensors.

1.2 Status and Note for usage

The DESY test beam facility includes three beam areas, TB21, TB22, with TB24 and TB24/1 in tandem as shown in Figure 2. **Two** Slow Control racks are currently available to move around at the test beam for different use cases.

Figure 2: Slow Control System Racks can be moved in T21/T22/T24 for observing.

The power and Internet connection should always be connected, while the Slow Control System is in use. Once the power was off, in order to continue the slow control data streaming, one needs to: 1) release the beam area interlock; 2) check to ensure the data logger is on, and the PC is on.

You can of course do the following things, however, please kindly inform the contact person **before**:

- disconnect the Ethernet cable and/or unpower the slow control system;
- replace the NTC sensors;
- relocate the rack by rolling it among different test beam areas or even to other work space;

2 DAQ architecture

2.1 Structure

The system data acquisition (DAQ) is implemented by two PCs running in parallel, see Figure 3. The DAQ is developped based on the current hardware, that user only needs to: 1) load a testbeam slow control *Producer* module in EUDAQ2 as you will do to load your device; 2) copy/paste the prepared slow control config file to your config file; 3) run the EUDAQ2 like how you will do to take data from your device. At the end of the run, once all the data is written to a standard EUDAQ2 raw file, you can find all the slow control data recorded as tags in your event.

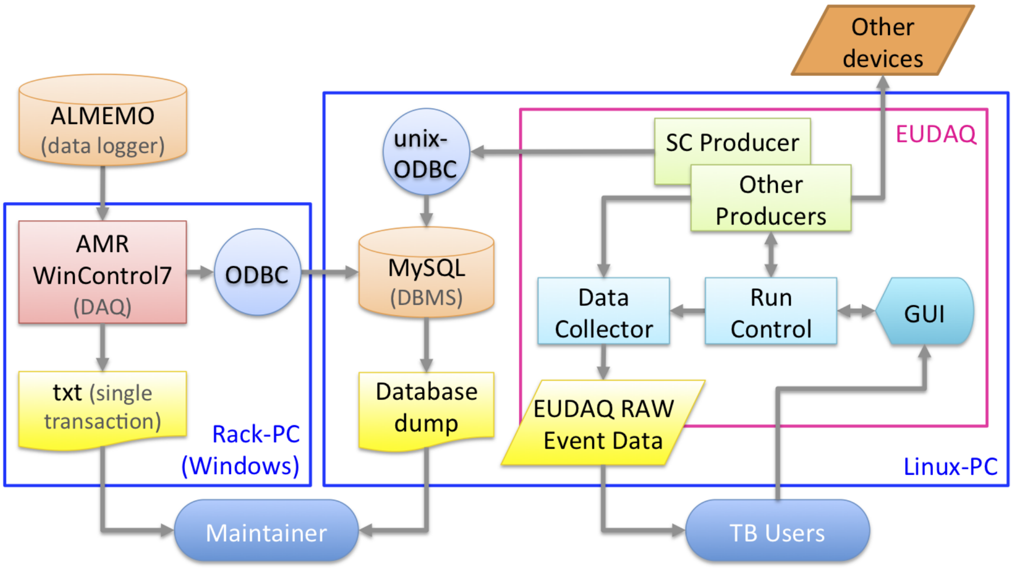


Figure 3: Flow chart of the Slow Control System

2.2 Windows Rack PC

The Windows PC on the rack collects the data from data logger (ALMEMO) with a commercial DAQ software (AMR). With a default (updated in Jan 2018) configuration: AMR software polls data from data logger every 30 s. An ODBC is set up to forward the values into a MySQL (version 5.7.20) based database on the Linux-PC.

2.3 Linux PC

The ODBC connector shown in Figure 3 connects the Windows Operating System (OS) with any remote DataBase (DB) using the DB's TCP/IP address. Therefore, if the internect connection of either of the two PCs is disconnected, no data can be transferred to the User.

A MySQL Database with necessary tables are set on a Linux PC at DESY, in order to transfer all the connected sensor channels; the relevant setup to send data out from the Windows PC is also prepared. As long as the Windows PC starts to Poll and send out data, the relevant tables in the MySQL DataBase will be filled with entries ordered by timestampes, if both PCs are on and connected to the Internet with their IP addresses not changed).

The EUDAQ2 software is installed on this Linux PC, and it reads out entries from the MySQL tables. A unix-ODBC is installed to connect the MySQL database with the EUDAQ2 framework. User can use the EUDAQ2 GUI with prepared Producer and Data collector modules loaded via a command line command, and once User start one EUDAQ2 run with all the relevant GUI, the slow control data will be written down to the standard EUDAQ raw event as tags. After data taking the output can be converted from EUDAQ raw/native file into a readable csv file, with a provided command line converter. Furthermore, there are macros provided to plot the csv data.

3 Software Setup

There are **two APIs** and **three softwares** to install and to configure:

- API:
  - ODBC: 32-bit needed, installed by default in Windows, need to install MySQL driver and configure a DSN for the remote MySQL DB;
  - unixODBC: installation needed, as well as a MySQL driver [2] to install and a DSN to add/configure for connection to MySQL DataBase;
- Software:
  - AMR: installed under purchase on the Windows Rack PC, ODBC DSN entry needed to add/configure for data transmit (Tx);
  - MySQL: installation needed, needs to add target DataBase with tables to receive (Rx) data sent from AMR;
  - EUDAQ2: installation needed, modules and configuration prepared for default setup; configurable variables in .conf files, to change DataBase and/or target tables.

3.1 MySQL

The MySQL database on the Linux PC is the central interaction point of the two DAQ softwares. First of all there has to be a user which can be used from multiple hostes. This is required so that the AMR software on the Windows PC can remotly access the database. Therefore the certain user has to get access privileges from different computers.

Installation

Install the MySQL server by using the Ubuntu package manage:

```
sudo apt-get updated
sudo apt-get install mysql-server
```

After the installtion, run the `mysql_secure_installation` utility, with which you can define the MySQL root user password, configure remote access and etc. It is simple to start the MySQL shell as the root user with the password you just set:

```
mysql -u root -p
mysql>
```

Remote access configuration

There are `database`, with `tables` inside, which are accessable for different `users` available at different `hosts` in MySQL.

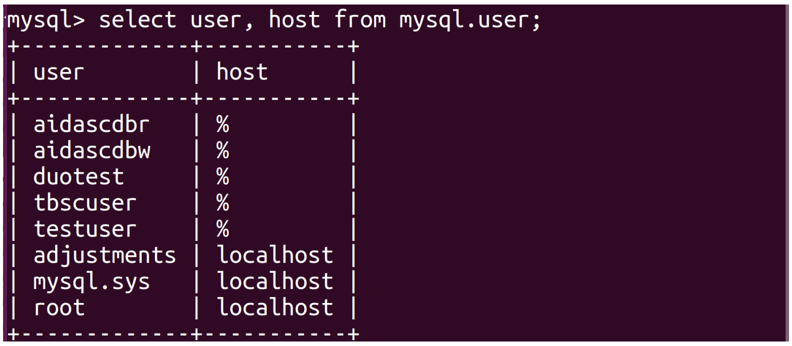
- At least two users needed: one is write-only for the AMR to access the DataBase and the other is read-only for the EUDAQ2.
- To maintain database/tables for different machines, one can use:
  - either: create one `database` for each rack, including one unit `table` and one data receiving `table`;
  - or: create one `database` for **ALL** racks, providing one common unit `table`, but one data receiving `table` for each machine.
  - the current strategy is to have one database for one rack with one unit tabke and one data receiving table provided.

Useful MySQL Commands

You can check all the users with the corresponding hosts using the following command:

```
mysql> SELECT user, host FROM mysql.user;
```

Then you should see an output as shown in Figure 4:



```
mysql> select user, host from mysql.user;
+-------------+-----------+
| user        | host      |
+-------------+-----------+
| aidascdbr   | %         |
| aidascdbw   | %         |
| duotest     | %         |
| tbscuser    | %         |
| testuser    | %         |
| adjustments | localhost |
| mysql.sys   | localhost |
| root        | localhost |
+-------------+-----------+
```

Figure 4: Check the access rights of which hosts to all the users in the MySQL, in this case, `aidascdbr` represents the read-only user prepared for EUDAQ2, while `aidascdbw` represents the write-only user prepared for AMR.

Notice: `%` is used as wild character, to allow the relevant useraccount to be accessed from all different hosts; for a certain user, it may `GRANT` such `PRIVILEGES` for different host to access different DataBases/Tables; to check it, one can do (eg. `aidascdbw`):

```
mysql> SHOW GRANTS FOR 'aidascdbw'@'%';
+---------------------------------------------+
| Grants for aidascdbw@%                      |
+---------------------------------------------+
| GRANT ALL PRIVILEGES ON *.* TO 'aidascdbr'@'%' |
+---------------------------------------------+
```

Currently, all privileges given to MySQL users registered for AMR and for EUDAQ2, to access from any host (`'aidascdbr'@'\%'`) to any database (`*.*`); once the privileges needed for each is fixed, one should change this. For a current setup, one can use the following commands to give privileges:

```
mysql> GRANT ALL PRIVILEGES ON *.* TO 'aidascdbr'@'\%' IDENTIFIED BY 'PASSWORD';
```

For creating a DataBase with tables, you can either log in any `USERNAME`, via which you want this database can be accessed; or you can log in as the `root` user, create the DataBase and later you `GRANT PRIVILEGES` to some `USERNAME`. Here are the example commands to create a DataBase named ‘tbscs’ with the a table named ‘tab_rx’:

```
mysql> CREATE DATABASE 'tbscs';
mysql> create table 'tab_rx' ('counter' int unsigned zerofill auto_increment, 'timer'
    datetime NOT NULL, 'temperature' double);
```

3.2 ODBC

This system is using the ODBC (Open DataBase Connectivity) in the Windows as interface between the AMRSoftware and the remote MySQL DataBase, and its 32bit version [4] is used due to compatibility which can be found in `c:\Windows\SysWOW64\odbcad32.exe`.

In order to talk to MySQL, a driver for the ODBC is needed, see Ref [2]. **Notice:** this was already done, so if user has any problem relevant (e.g. different DataBase needed and etc.), please contact the system admin as only the administrator of the Windows PC is allowed to do this.

For each MySQL DataBase, it needs to be registered as a System Data Source Name (DSN) or User DSN for AMR to recognize. Here is an simple example to show how to add a new DSN, please notice, only administrator can add a system DSN, others can only add a user DSN. It is tested in Januaray 2018 with our first user experience, that AMR can work with a user DSN without any problem.

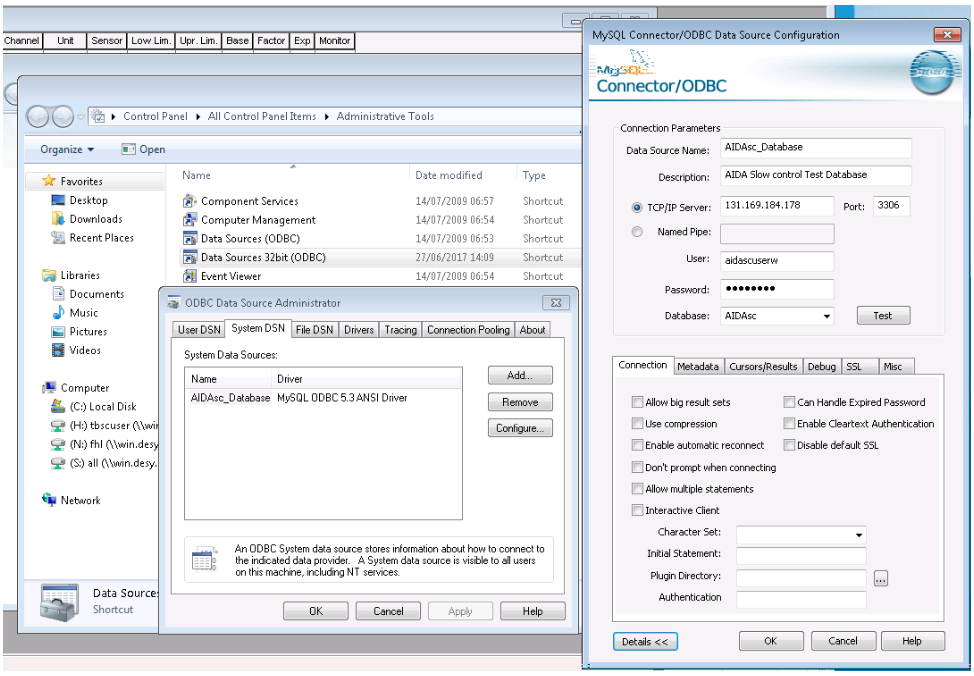


Figure 5: The windows one needs to open to register a new System/User DSN for connecting a MySQL DataBase to a Windows system via ODBC.

The windows one needs to open to register a (user) DSN is shown in Figure 5:

- Open the ODBC data source administrator: `c:\Windows\SysWOW64\odbcad32.exe`;
- Click ‘Add’ under the register User or System/User DSN, then choose the ‘MySQL’ driver and continue with the configuration window (rightmost in Figure 5).

- Under the 'MySQL Connector/ODBC Data Source Configuration' window: name your data source, give the correct TCP/IP address of the PC where the DataBase is located, Port is usually given as 3306, and put the User with its password to access the Database you are going to register. Please **notice** that you give this Windows PC access right to under this MySQL User to the DataBase you are going to use.

- Choose the target DataBase for transferring slow control data. To test if your connection is working or not, simply click on the 'Test' button.

3.3 AMR

The AMRSoftware was installed under purchase form the ALMEMO company. If there is a problem encountered, please contact the local support group, and we can also get support by the company. If there is any new hardware added, please also contact the local support group for updating with the AMR software.

One needs to configure the AMR in terms of online export via ODBC, to do so, one simply needs to click `AutoSave` from the AMR Software window, then click `Add...` and choose `Add Online ODBC Export` as shown in Figure 6 (left). Also give one example to show how the DSN is added for AMR to talk with, and how to start it proper to transfer data.

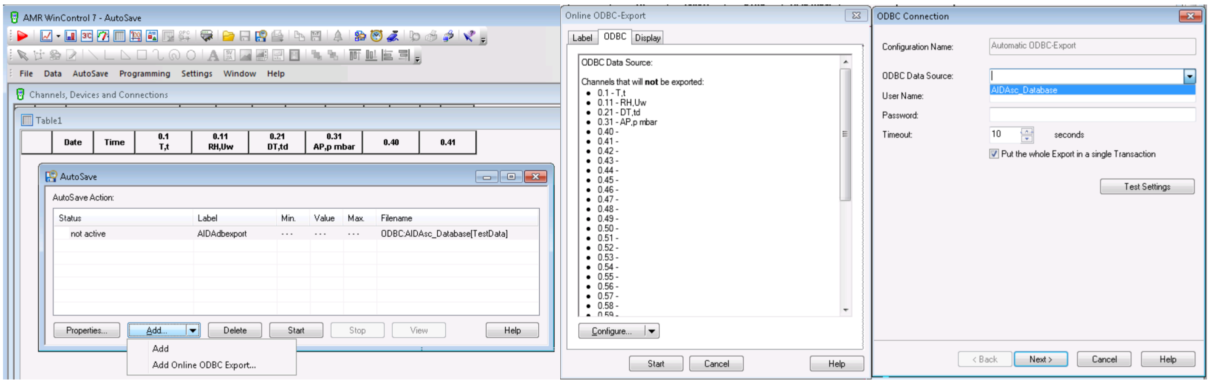


Figure 6: Windows for adding an online ODBC Export entry to AMR software for exporting data out to a remote/local database via ODBC.

3.4 unixODBC

On the Linux PC where the EUDAQ2 is installed, a linux version of ODBC, namely unixODBC, is used. One can of course use the MySQL C++ connector library to connect MySQL DataBase with C++ code directly, but the point here is to provide a flexible scheme for this system to move to different SQL DataBase easily.

The unixODBC is not installed by default in linux, and it can be download from http://www.unixodbc.org/. A driver is also needed to connect to a MySQL DataBase, that can be downloaded from https://dev.mysql.com/downloads/connector/odbc/. Drivers are not automatically added to unixODBC, one also needs to configure it. All of the installation should be done already, in case of any change needed, please contact the system admin for administrator right to modify/change.

The unixODBC will be initialized everytime you login the system, using a init file `/etc/odbcinst.ini`. This file is already moved to `/usr/local/etc` in order to provide access of the DSN to all the linux Users instead of admin only. Therefore, please add the following line to your shell init file, such as `.bashrc`:

```
export ODBCSYSINI=/usr/local/etc
```

Add a DSN

One can create a DSN template file, `/usr/local/etc/odbc.ini`, as shown in Figure 7 then install/register it to the system. An example to show how-to is followed:

(i) In the `odbc.ini`, one needs to start a DSN with its name, as 'odbcAidaSC' in Figure 7, then specify the DataBase, username with password, DataBase server and unixODBC Driver. It is not of importance where the template file is stored.

(ii) Once a `odbc.ini` is newly added or updated, run the `odbcinst` executable to register/update with the following command in the terminal [3]:

```
$ odbcinst -i -s -f /usr/local/etc/odbc.ini
```

To check if the DSNs are added correctly to the system via unixODBC:

```
$ odbcinst -q -s
[odbcAidaSC]
```

3.5 EUDAQ2

The EUDAQ2 [1] is the second version of the EUDAQ data acquisition framework, which is commonly used for many lively running Mimosa telescopes in the world. At DESY, there are two Mimosa telescopes, Duranta and Datura that are frequently demanded by users (70% users in 2017), and one AIDA2020 strip telescope under construction, all of which are using EUDAQ2 as the DAQ user interface. Therefore, to unify the DAQ user interface for test beam users at DESY, this system is also designed based on the EUDAQ2 framework.

The EUDAQ2 software is usually provided and maintained by the local testbeam support group. In case of need, one can find simple instruction to download and install an EUDAQ locally from here: http://eudaq.github.io. **Notice:** please be sure that you have `cmake` and `cmake-qt-gui` installed to compile the code, if not, use `sudo apt-get install` command to install them; a `Qt5` is also required to have the EUDAQ2 GUI produced, if not please download an open source version of the Qt Framework from their website: https://www.qt.io.

Notice: the EUDAQ2 modules for this sytem is not yet merged into the central master branch, for testbeam users, it should be already provided on the hut PC. In case that one needs to install on his/her own, please follow the following steps:

- once you have the EUDAQ2 master branch installed from the central git repository, go to `$EUDAQ2/user/` directory;

- clone the slow control modules here:

```
$ git clone https://github.com/DESY-TBSC/DESYtbsc.git
```

- go to `$EUDAQ2/build/` directory, do `cmake-gui ..`, and make sure you have options with `USER_TBSCDESY` all chosen under the `USER` category;

- compile it with all executables/librairies installed:

```
$ make install -j4
```

```
[odbcAidaSC]
Description=AidaSC Database
Driver=MySQL ODBC 5.3 Driver ANSI
SERVER=localhost
USER=
DATABASE=larsAida
PORT=3306
SOCKET=/var/run/mysqld/mysqld.sock
INITSTMT=
CHARSET=
PREFETCH=
READTIMEOUT=
WRITETIMEOUT=
SSLCA=
SSLCAPATH=
SSLCERT=
SSLKEY=
SSLCIPHER=
SSLVERIFY=
FOUND_ROWS=
BIG_PACKETS=
NO_PROMPT=
DYNAMIC_CURSOR=
NO_SCHEMA=
NO_DEFAULT_CURSOR=
NO_LOCALE=
PAD_SPACE=
FULL_COLUMN_NAMES=
COMPRESSED_PROTO=
IGNORE_SPACE=
NO_BIGINT=
NO_CATALOG=
USE_MYCNF=
SAFE=
NO_TRANSACTIONS=
LOG_QUERY=
NO_CACHE=
FORWARD_CURSOR=
AUTO_RECONNECT=
AUTO_IS_NULL=
ZERO_DATE_TO_MIN=
MIN_DATE_TO_ZERO=
MULTI_STATEMENTS=
COLUMN_SIZE_S32=
NO_BINARY_RESULT=
DFLT_BIGINT_BIND_STR=
NO_INFORMATION_SCHEMA=
INTERACTIVE=
CAN_HANDLE_EXP_PWD=
ENABLE_CLEARTEXT_PLUGIN=
NO_SSPS=
UID=tbscuser
PASSWORD=DESY2tbsc
TraceFile=/tmp/odbc.trace
Trace=1
```

Figure 7: An example of a unixODBC DSN template file: `odbc.ini`.

4 How-to Take Data

Once all the APIs and Softwares from Secion 3 are well installed and configured. To prepare for data taking: firstly one needs to make sure the data logger and the Windows PC on the rack are on, as well as the Linux PC where the MySQL and EUDAQ2 are installed is on; seconldy, both PCs need to connect to the internet with the same IP addresses as registered in the DSNs, and in the server/host access privileges given to the MySQL users. Any unexpected errors may arise if the system is not well configured/prepared. Before operating any data taking from the EUDAQ2 side, one needs to start the `polling` from the AMR system, and make sure that, in the AMR software, the `Online ODBC Export` entry under the `AutoSave` is 'Start'.

Experience from the first user experience in January 2018 shows that the most common errors usually come from: wrong connections in the conf file and/or the AMR software is not polling and/or the ODBC export process is not 'Start'.

Here below shows one step-by-step example, about how-to get slow control system data polled into your own EUDAQ2 data stream (integrated into your event as a tag):

(i) **Operate the AMR software**

a) remote log into the Windows PC on the rack using the `xfreerdp`, e.g.:

```
xfreerdp /w:1600 /h:1000 /u:tbscuser /v:fhl-tb-sc01 /cert-ignore
```

b) open the AMR software from the Desktop, as shown in Figure 8;

c) verify the ODBC connection is active (in case the MySQL PC is not connected or configuration went wrong), by clicking ‘Test’ in the ‘MySQL Connector/ODBC Data Source Configuratio’ window, shown in bottom left in Figure 8;

d) **START ODBC Transfer:** in the AMR software window, open `AutoSave`, choose the entry with ‘export’ lablled, and click `Start`;

e) **START DATA POLLING:** click the RED arrow bottom on the top left of the AMR software window, see Figure 8, to start the `Polling` process, and it will start blinking red and yellow when data is taken;

f) **STOP DATA POLLING:** click the the same RED bottom to stop.

(ii) **Start from EUDAQ2**

a) go to your EUDAQ2 directory `cd $EUDAQ2`;

b) to start the EUDAQ2 GUI:

```
cd bin/
./euRun
```

c) to load the slow control modules, Producer to connect to the MySQL, and DataCollector to write data into a EUDAQ2 raw file:

```
./euCliProducer -n tbscProducer -t tbsc
./euCliCollector -n tbscDataCollector -t tbscDC
```

d) all the loaded modules will be shown in the `Connections` panel, see Figure 9, then to load the init and config files, simply click ‘Load’ button. Examples init/conf files are available in the directory of `$EUDAQ2/user/tbscDESY/misc/`.

5 How-to display results

An online monitor is currently under development, in collaboration with the DQM4Hep team, as a sub-task of integrating DQM4Hep to EUDAQ2. Besides, two methods provided to display the slow control results after one data taking run:

- Either one can use `euCliReader` under the `$EUDAQ2/bin/` directory, to readout all the events recorded in the `.raw` output;

- Or one can use `tbscCliConverter` under the `$EUDAQ2/bin/` directory, to convert the slow control variables of all the events from the `.raw` file to a `.csv` file, see command example:

```
./tbscCliConverter -i tbsc02_180117123547_run000070.raw -o test_out.csv -ip
```

  The option `-ip` is optional, it is to enable event printing during the converting process.

Histograms of temperature and humidity collected by different sensors on the rack are shown in Figure 10, based on data collected in testbeam area T24 at DESY. A nice agreement is expected to observe from data output from EUDAQ2 to data from MySQL databse dump, which validate the system stability.

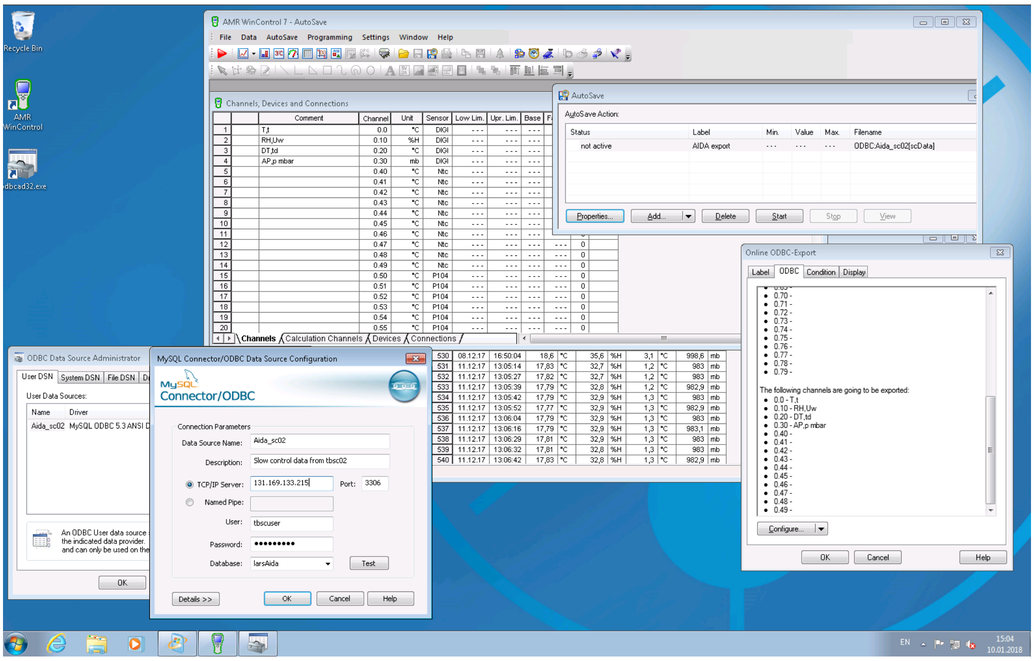


Figure 8: The Windows OS from the rack PC via a xfreerdp, with opened windows of the AMR software (AMR WinControl7, top central) with its AutoSave configuration (right), and ODBC Data Source Adminitration with MySQL connector configuration (bottom left).

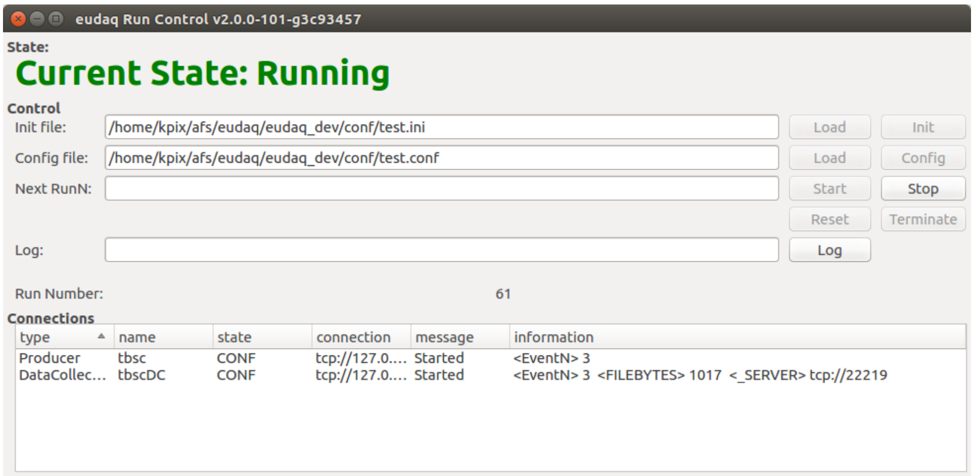


Figure 9: EUDAQ2 GUI example with slow control system modules loaded, example init and conf files loaded.

6 Integrate your own DataBase

The current target is to integrate at least one user application to this system.

- User can either integrate easily by writing a small script to send their data to any favored DataBase.

- To integrate a different structured DataBase from user's monitor to the EUDAQ2, users can either choose to do their own or contact the local support group for help. This process can be a quite quick (1-2 workdays) development because of using the `unixODBC`.

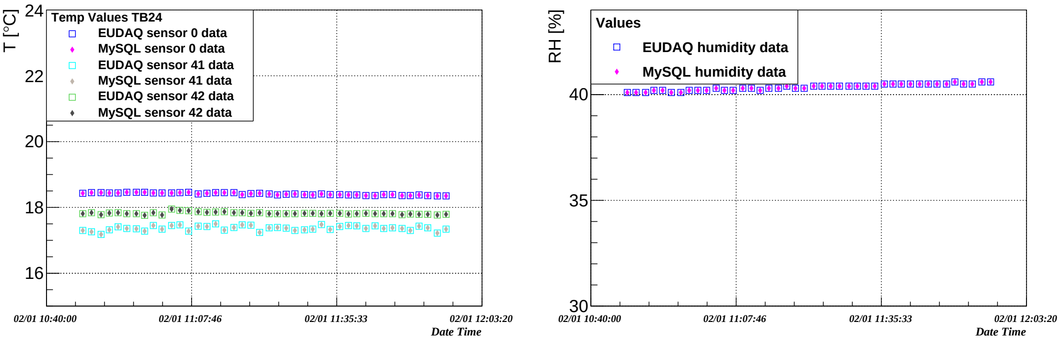


Figure 10: Histograms of temperatures (left) and humidity (right) collected by slow control system sensors in January 2018. The empty marks show data collected via EUDAQ2, and the filled marks come from MySQL data, expected agreement observed.

An example may be given with a later update of this document, or one can check the developer manual:

https://cds.cern.ch/record/2284369/files/AIDA-2020-NOTE-2017-007.pdf

References

[1] Wing, M. et al., *Development of run control ready*, AIDA2020, AIDA-2020-MS62, http://cds.cern.ch/record/2276456

[2] https://dev.mysql.com/downloads/connector/odbc/

[3] https://manpages.ubuntu.com/manpages/xenial/man1/odbcinst.1.html

[4] https://docs.microsoft.com/en-us/sql/odbc/microsoft-open-database-connectivity-odbc?view=sql-server-2017