MicroPMU Quick Start Kit

Administrator Manual

Revision 9

Table of Contents

1 Quick Start Server Startup/Shutdown

Before getting started with the MicroPMU setup, the Quick Start Kit server needs to be powered up. To power your server, first make sure it is plugged into a suitable power supply and then press the button directly to the right of the Dell logo which can be found at the top left corner of the server's front panel.

IMPORTANT: Please note that this button should not be used to force a server shutdown, nor should the power cable be unplugged while the server is powered on. Please refer to the next section on how to properly shutdown your server.

Upon pressing the power button, a green light should be illuminating the button and there should be audible and visible cues that the server began the boot process. While the server is booting, some options may be available for the F1, F2, F3 keys, etc. Do not press any of these keys, and let the server continue to boot until the login prompt is displayed.

1.1 Quick Start Server Login

Once the server is ready the prompt will show…

```
Login as:
```

The default Quick Start Kit server user is: manager

Now the prompt will show…

```
Password:
```

The default quick start server password is: Company*2016!

Upon a successful login the prompt will show…

```
manager@quickstart:~$
```

1.2 Quick Start Server Logout

To log out of a user account or ssh session, simply type logout in the command prompt and press the enter key.

```
logout
```

1.3 Quick Start Server Shutdown

It's important to try to gracefully shutdown the quick start server whenever a shutdown is required. To do this, first make sure manager is logged in, then type shutdown now into the command prompt and press the enter key.

```
shutdown now
```

2 Quick Start Server Remote Login Using SSH

SSH allows logins to a remote computer/server over an encrypted connection. To SSH into the Quick Start Kit server, either an SSH client for Windows (we recommend PuTTY) is required, or the terminal can be used on a Mac or Linux computer.

2.1 Finding Your Quick Start Server IP

Before attempting to establish an SSH connection to your quick start server, the server IP address must be noted.

By default, the Quick Start Kit server IP address is set to 172.16.1.100.

To find or double check the IP, login as manager on the server itself (using a monitor and keyboard plugged into the physical server). Enter the following command to display the server IP.

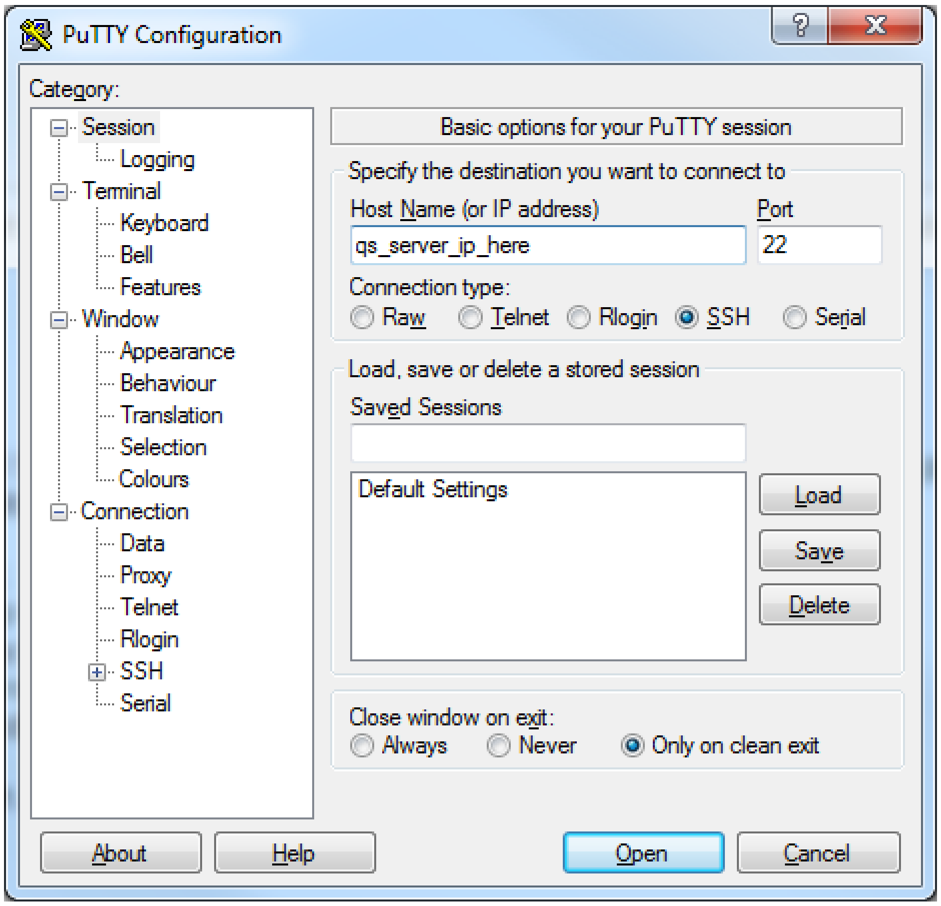
```
ifconfig | grep "inet addr"
```

Make sure to type the command exactly as seen and at least two lines of text that start with "inet addr" should be printed on the screen. Take note of the "inet addr" on the line that also displays "Bcast" and "Mask", this is the server IP address.

2.2 SSH Using Windows

To SSH from a Windows computer, a third party SSH client will need to be installed. We recommend using PuTTY as it is reliable, safe, easy to use and free! If PuTTY is not downloaded already, the program may be downloaded from www.putty.org.

Open PuTTY and the PuTTY Configuration window should be displayed. To start an SSH connection enter the Quick Start Kit server's IP address as well as make sure the port is set to 22 and the connection type is set to SSH.



Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): qs_server_ip_here
Port: 22

Connection type: Raw Telnet Rlogin [✓ SSH] Serial

Load, save or delete a stored session

Saved Sessions

Default Settings

Close window on exit: Always Never [✓ Only on clean exit]

This configuration can be saved by typing in a configuration name under "Saved Sessions" and click the save button. From there, just highlight the configuration name and click the load button then click the open button to start a connection.

A prompt may ask if host to connect to is trusted. Type and enter "yes" to finish the connection process. Once connected, proceed to login the same way as on the server itself (Login as: manager Password: Company*2016!).

2.3 SSH Using Mac OSX

To SSH from a Mac, there is no need to use third party software. Simply open a terminal window which can be found by using the spotlight search (press "command" + "space" or click magnifying glass icon toward the top right corner of your screen) and typing "terminal". Enter the following command in the terminal to establish an SSH connection with the quick start server.

```
ssh manager@quick-start-server-ip
```

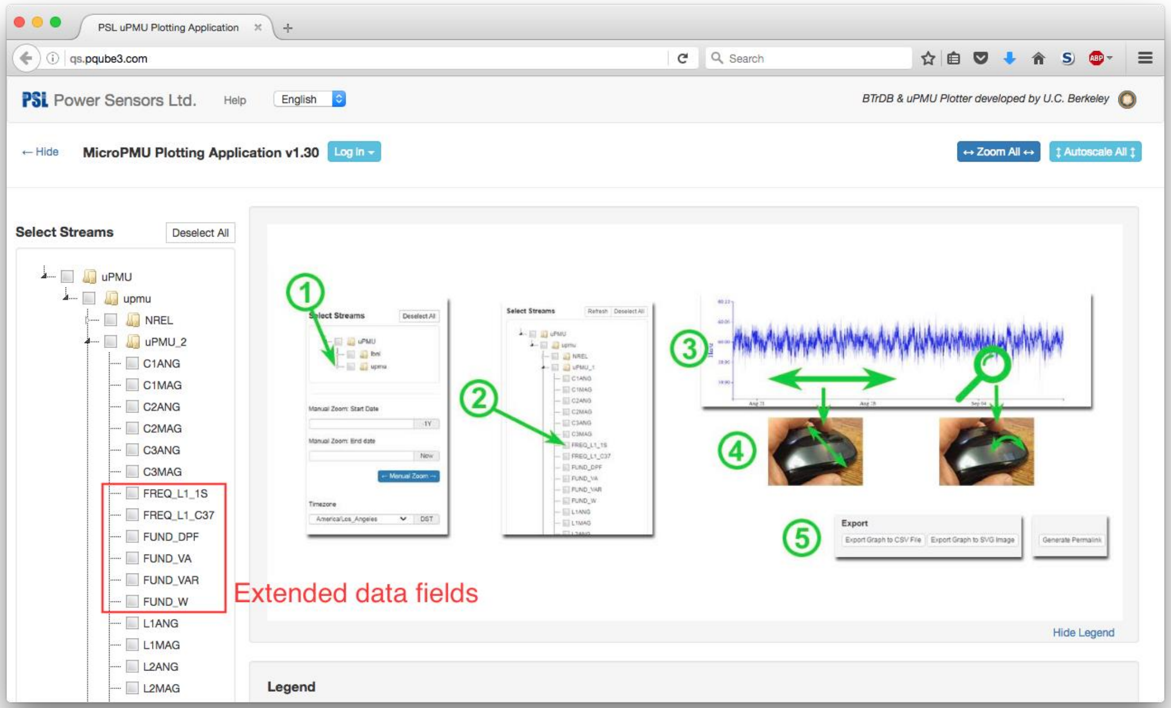
Example:

```
ssh manager@172.16.1.100
```

A prompt may ask if host to connect to is trusted. Type and enter “yes” to finish the connection process. Once connected, proceed to login the same way as on the server itself (Password: Company*2016!).

3 Adding MicroPMUs

MicroPMUs have two .dat file modes: base and extended. The difference between these two modes is the amount of data fields that are recorded. The extended .dat file mode will present frequency, apparent power, real power, and power factor data in addition to the base data fields as can be seen in the screenshot below of the MicroPMU Plotter website. Base mode will generate files around 379KB in size and extended mode will generate files around 552KB in size.



To add a MicroPMU, enter the following commands:

```
cd
```
(to go to the home directory if you are not already there)

```
./upmu-adm
```

(run upmu-administration panel)

add **<u>ipaddr</u>** **<u>serial</u>** **<u>alias</u>** **[ext]**
(upmu ip address, upmu serial number, upmu alias name, optional extended mode)

Note that any name can be used for upmu alias. If the **`ext`** parameter is not included, .dat files will default to base mode.
To add a MicroPMU in base mode, simply enter the **`add`** command without the **`ext`** parameter.

Example:

```
add 192.168.1.24 P3112233 upmu_1
```

To add a MicroPMU in extended mode, enter **`true`** at the end of the **`add`** command.

Example:

```
add 192.168.1.24 P3112233 upmu_1 true
```

3.1 Finalizing Added MicroPMU(s)

`list`
(view list of MicroPMUs, their IP address, extended mode status and aliases to confirm changes)

`save`
(you must type save to write to <u>upmuconfig.ini</u> file. otherwise, your changes will not be saved)

`Ctrl+C`
(to exit upmu-adm)

Upon a successful exit, the command prompt should print a message like “clean exit – no unsaved changes”.

Next, update the metadata database:

`cd`
(redirects you to the home directory)

`python manager2lite.ipy`
(adds metadata or updates existing plotter metadata)

Status messages will confirm that existing metadata have been added, removed, or updated.

Finally, restart services:

```
sudo supervisorctl restart all
```
(enter manager password when prompted)

Note the services will take up to 30 seconds or less to stop and start again. Any "spawn error" messages from this command may be ignored.

3.2 Switching Between Base and Extended Mode

To switch between base and extended mode, first make sure **`upmu-adm`** is running. This can be run from the **`/home/manager`** directory with the **`./upmu-adm`** command. Enter the following command to switch the .dat file mode (true=extended, false=base).

```
setext ipadrr [true | false]
```

Example:

```
setext 192.168.1.101 true
```

Note that the base mode data file size will be around 379KB and the extended mode file size will be around 552KB.

4 Removing MicroPMUs and MicroPMU Data

To remove an existing MicroPMU, the metadata database will also need to be updated.

First, login to the Quick Start Kit server as manager, then enter the following commands.

```
cd
```
(switches to home directory if not already there)

```
./upmu-adm
```
(starts MicroPMU administration panel)

```
list
```
(view list of MicroPMUs, their IP address, serial number, extended mode status and aliases)

```
rem [IP | Serial | Alias]
```
(remove the MicroPMU by specifying one of the following: IP Address, Serial Number, Alias)

Example:

`rem 192.168.1.24`
(remove MicroPMU with the entered IP address)
`rem P3001234`
(remove MicroPMU with the entered serial number)
`rem upmu_1`
(remove MicroPMU with the entered alias)

`save`
(save changes)

`Ctrl+C`
(exit)

Upon a successful exit, the command prompt should print a message like “clean exit – no unsaved changes”.

`cd`
(redirects to the home directory)

`python manager2lite.ipy`
(removes metadata or updates existing plotter metadata)

A message will print stating that metadata is being removed for the MicroPMU chosen to remove along with processing messages for any other MicroPMUs available.

Finally, restart services:

`sudo supervisorctl restart all`
(enter manager password when prompted)

Note the services will take up to 30 seconds or less to stop and start again. Any “spawn error” messages from this command may be ignored.

5 Importing Existing .dat Files

Existing data files can be imported to **plotter** by using **`upmu-adm`** command-line interface.

NOTE: The MicroPMU that the .dat files belong to must be added to the plotter instance on the server the data will be imported to. For example, say data files will be imported for a MicroPMU with a serial number of P3001234. If a MicroPMU with this serial number has not been added by using the **`upmu-adm`** program, then it will need to be added first. Please refer to the “Adding MicroPMUs” section for instructions on adding a MicroPMU to the plotter.

First the data files need to be present on the Quick Start server. If there are already data files on the server then proceed to the importing data step.

5.1 Copying Data Files to The Server

5.1.1 Copy Data Files via SCP

Otherwise if data files are stored on a remote computer then the **`scp`** command can be used to move them to the server (Linux and Mac OSX only).

First create a directory on the server to copy the data files into.

```
cd
mkdir data-import
```

Now, from the remote computer, make sure the data files to import are located in a folder and then use the following command on the command line of the remote computer.

```
scp -r /path-to-folder-containing-data-files/
manager@172.16.1.100:/home/manager/data-import/
```

(replace example IP with the Quick Start server IP and use a space between the local path and the Quick Start server path)

5.1.2 Copy Data Files via USB/External Drive

Alternatively, a FAT32 formatted USB or external drive may be plugged directly into the server to copy the .dat files.

Plug the external drive into the Quick Start server via USB port.

NOTE: After plugging in the external drive, a message may appear displaying something to the effect of “No Caching mode page found…Assuming drive cache: write through”. Ignore this message and use **`Ctrl+c`** in order to make the command prompt appear again.

Now use the following command to list the available drives that the server sees.

```
sudo fdisk -l | grep sd
```

In the list should be some entries for “Disk” that will list a path such as **`/dev/sda/`** or **`/dev/sdb/`**. Those are the two drives installed in the server. If there are no other drives installed in the server then the external drive will likely be listed as **`/dev/sdc/`**. Confirm which drive is the external drive by checking the size listed as well as the type, which should include FAT32. Underneath the “Disk” line should be a list of partitions for the drive such as /dev/sdc1, /dev/sdc2, etc. depending on how many partitions (independent file systems) are on the drive which will most likely just be one partition unless explicitly formatted otherwise.

Once the path of the external drive is found, mount the drive with the following commands (assuming the drive is located at **`/dev/sdc/`** and only has one partition).

Example:

```
sudo mkdir /media/external
```
(if the directory does not already exist)

```
mount -t vfat /dev/sdc1 /media/external
```

5.2 Importing Data

Now everything is staged to begin the data import.

To access **`importdat`**, type the following commands:

```
cd
```
(go to home directory)

```
./upmu-adm
```
(to run upmu-administration panel)

```
importdat FULL-PATH-TO-DIRECTORY
```
*(where "FULL_PATH_TO_DIRECTORY" is where the .dat files are located. Copy ALL .dat files to import into the directory. The process is not recursive, so all files must be in the directory. For help, type **`importdat`** with no arguments for usage instructions or **`help`** to show all commands)*

Example (if data is stored in a "data" folder on the external drive):

```
importdat /media/external/data
```

Then:

```
Ctrl+C
```
(exit)

Upon a successful exit, the command prompt should print a message like "clean exit – no unsaved changes".

Once finished with a data import, unmount the external drive with the following command.

```
sudo umount /media/external
```

Afterwards, wait up to 180 seconds to let the **sync2_quasar** service automatically add the data to the plotter instance (sync2_quasar runs once every 180 seconds). Remember that files that have been imported are either in "base" or "extended" mode.

6 Backing Up And Restoring MicroPMU Databases

Databases can be manually backed up by copying the entire **`/ssd/data/`** and **`/ssd/srv/`** directories to a safe location, but first, please read the tips and instructions below to ensure the backup will be done safely.

6.1 Important Notes for A Database Backup/Restore

They may become quite large, so the **`/ssd/data/`** and **`/ssd/srv/`** directories may be backed up periodically either remotely or by physically backing up the storage drive. Be careful when using RAID setups, as corrupt databases may automatically overwrite working copies on other drives.

It is best practice to save the directories while there are no services running. This way, all data leading up to the backup date will be preserved when the files are copied back onto the new drive, and queued data will begin piping in from MicroPMUs.

Do not remove the first drive, which contains the operating system and plotter settings. Remove the secondary storage drive from the server and mirror its contents to the new drive.

How to avoid corrupting your database:

1. **IMPORTANT:** never hot unplug the Quick Start server while data is being synchronized or written.
2. **IMPORTANT:** make sure to stop all services during backup so new data isn't being written:

   ```
   sudo supervisorctl stop all
   ```

3. **NOTE:** Hooks to the operating system exist to safely shut down scripts on graceful reboot or shutdown. The following commands are **SAFE** for powering down the server to do a physical drive backup.

   ```
   sudo reboot now
   sudo shutdown now
   ```

6.2 Example Backup/Restore

As mentioned earlier, the databases can be backed up by copying the contents of `/ssd/data/` and `/ssd/srv/` to a safe location, by mirroring the physical drives, or by copying the drive contents over a network connection to a remote drive.

The most reliable, safest and simplest means of backing up the data is to directly connect/plug in an external drive to the Quick Start Kit server and copy the data to that drive. To do this, start by acquiring an external drive (or USB drive, granted it will have enough space) formatted to FAT32 and make sure it has a USB port and that it will have enough space to hold the backup copies. Refer to the plotter website to see how much disk space is currently occupied on the server.

Plug the external drive into the Quick Start Kit server via USB port.
NOTE: After plugging in the external drive, a message may appear displaying something to the effect of "No Caching mode page found…Assuming drive cache: write through". Ignore this message and use `Ctrl+c` in order to make the command prompt appear again.

Now use the following command to list the available drives that the server sees.

```
sudo fdisk –l | grep sd
```

In the list should be some entries for "Disk" that will list a path such as `/dev/sda/` or `/dev/sdb/`. Those are the two drives installed in the server. If there are no other drives installed in the server then the external drive will likely be listed as `/dev/sdc/`. Confirm which drive is the external drive by checking the size listed as well as the type, which should include FAT32. Underneath the "Disk" line should be a list of partitions for the drive such as /dev/sdc1, /dev/sdc2, etc. depending on how many partitions (independent file systems) are on the drive which will most likely just be one partition unless explicitly formatted otherwise.

Once the path of the external drive is found, mount the drive with the following commands (assuming the drive is located at `/dev/sdc/` and only has one partition).

```
sudo mkdir /media/external
mount -t vfat /dev/sdc1 /media/external
```

Now everything is staged to begin the backup, but first remember to stop all services!

```
sudo supervisorctl stop all
```

Now backup the data with the following commands.

```
sudo cp –r /ssd/data /media/external
sudo cp –r /ssd/srv /media/external
```

Note that data backups can be restored to the server by switching the placement in the paths like so. Make sure the ssd directory is empty before copying files to it.

```
sudo cp –r /media/external/data /ssd
sudo cp –r /media/external/srv /ssd
```

Once finished with a backup or restore, unmount the external drive with the following command.

```
sudo umount /media/external
```

Now services may be started back up.

```
sudo supervisorctl start all
```

Note the services will take up to 30 seconds or less to start again. Any “spawn error” messages from this command may be ignored.

7 Deleting Data and Starting Fresh with a New Instance

Take note before deleting database data:

IMPORTANT: Remember to copy the entire **/ssd/data/** and **/ssd/srv/** directories and `upmuconfig.ini` to a safe location in case you need to restore it to an old state.

IMPORTANT: never hot unplug the Quick Start server while data is being synchronized.

IMPORTANT: make sure to stop all services with the following command:

```
sudo supervisorctl stop all
```

The file-based database can be deleted to start a new instance by clearing the contents of `/ssd/data/db/` and `/ssd/srv/data/db`.

To clear these directories, type the following commands:

IMPORTANT: Type these commands exactly as seen. Deleting the wrong directory (such as `/` by itself, which is the root directory) could potentially brick the server.

```
sudo rm -rf /ssd/data/db/
sudo mkdir /ssd/data/db/
sudo rm -rf /ssd/srv/data/db/
sudo mkdir /ssd/srv/data/db/
```

Run the following command to start **mongo** and create a new database instance**.**

```
sudo supervisorctl start mongo
```

Run these commands to create a new **btrdb** instance.

```
cd /home/manager/go/bin
sudo ./btrdbd --makedb
```

(this creates another instance of BTrDB)

Now, start the **btrdb** service**:**

```
sudo supervisorctl start btrdb
```

Remove **upmuconfig.ini** and **backupconfig.ini** in **`/home/manager/`** directory with the **`rm`** command.

```
cd
rm upmuconfig.ini
rm backupconfig.ini
```

New MicroPMUs may now be added. Refer to the section that covers the process for adding MicroPMUs and go through all the steps to properly add the MicroPMUs.

8 Setting a New Quick Start Kit Server IP Address

Open the network interfaces file for editing with the following command.

```
sudo nano /etc/network/interfaces
```

(Opens the file in the nano text editor. Alternatively, the vi/vim text editor may be used. Nano is recommend as a more user friendly text editor if it is unclear which one to use.)

Edit the highlighted sections with the desired static settings:

```
auto eno1
iface eno1 inet static
      address 172.16.1.100
      netmask 255.255.255.0
      gateway 172.16.1.1
      #dns-nameservers XXX.XXX.XXX.XXX
      #dns-nameservers XXX.XXX.XXX.XXX
```

(DNS nameservers are optional, so they are by default commented out. Nameservers are used for assigning domain names to IP Addresses).

Save changes in **`nano`** with **`Ctrl+x`** then type in the letter "y".

Restart the network interface with the following command.

```
sudo ifdown eno1 && sudo ifup eno1
```

(The server can be restarted with the command, **`sudo reboot now`**, to apply new settings)

NOTE: Automatic dhcp settings should not be used because MicroPMUs need a static IP to send the data to. The dynamic settings are available in the network configuration file if a dynamic IP address needs to be assigned to the server.

9 Routing MicroPMU data to the Quickstart Kit Server

The default IP Address for the Quick Start Kit server is 172.16.1.100
If this has been changed, the MicroPMU setup.ini files will need to be updated with the new IP address. Change this variable in the setup.ini file of each MicroPMU that will send data to the Quick Start Kit server:

```
;-----------------------------------------------------
[microPMU_Database_Server_Settings]
;-----------------------------------------------------
; ------ If you have a microPMU database server, enable this setting
; ------ to automatically push data from your microPMU to the server.
; ------ Valid values: ON, OFF
Enable_Push_Data_to_Database_Server=ON
Database_Server_IP_Address=172.16.1.100
```

Once complete, restart the MicroPMU for changes to take effect. Make sure that the MicroPMUs are on the same domain as the Quick Start Kit server if they are on a closed network with the provided Ethernet switch.

```
;-----------------------------------------------------
[Network_Setup]
;-----------------------------------------------------
; ------ Valid Values: Use_DHCP, Use_Fixed_IP
IP_Address_Method=Use_Fixed_IP
Publish_IP_Address=ON

;-----------------------------------------------------
[Fixed_IP]
;-----------------------------------------------------
; ------ This section is ignored if the IP_Address_Method is set to
Use_DHCP
```

```
IP_Address=172.16.1.101
IP_Gateway=172.16.1.1
IP_Mask=255.255.255.0
IP_DNS1=8.8.8.8
IP_DNS2=8.8.4.4
```

NOTE: The serial number of each MicroPMU **MUST MATCH** the serial number entered in the **`upmuconfig.ini`** file (this can be checked with **upmu-adm's list command** or view the contents of **`upmuconfig.ini`** in a text editor). If the serial number ever changes, update this in **`upmuconfig.ini`**. The **receiver** process validates and sorts data by serial number, so it will reject data from unknown serial numbers.

10 Changing the Root User Password

By default, the username for the MicroPMU Quick Start server is manager. You are also provided with a default manager/root user password, although it is recommended to change it for increased security. To change it, type:

```
sudo passwd
```

From there, follow the instructions to change the root password. This should mean entering the existing password, followed by the new password twice. Also note that the password will not be displayed as it is entered so be sure to type with precision. Be sure to take note of the new password and record it in a safe place!

11 Changing MicroPMU Domains for Access Control

Edit **`/home/manager/upmuconfig.ini`** and change the domain of the MicroPMUs to a separate domain (for example from **`upmu`** to **`lbnl`**). In **`upmuconfig.ini`**, find the **`Path`** fields for the MicroPMUs to assign to new domain(s). Each **`Path`** field for each unit will need to be changed.

```
Path=/upmu/upmu_name/stream_name
```

becomes:

```
Path=/lbnl/upmu_name/stream_name
```

After saving the changes to **`/home/manager/upmuconfig.ini`**, re-run **`manager2lite.ipy`**

Changes will not show until made public. The domain needs to be made public by editing **`/home/manager/go/src/github.com/SoftwareDefinedBuildings/mr-plotter/tools/tagconfig.json`** and assigning the domain name (ex. **`lbnl`**).

All units in a domain can be made public by using an open ended domain like **`/upmu`** or **`/lbnl`** or a combination of devices from different domains. The existing file has **`"public": ["/upmu"]`**, so all devices in **`upmu`** domain are open to the public by default. To make all domains public, make **`“public”: [“/”].`**

Syntax for singular units made public:

```
{
      "public": ["/lbnl/alameda_conference1","/upmu/san_diego"],
}
```

Syntax for multiple units in a domain made public:

```
{
      "public": ["/lbnl", "/upmu"],
}
```

Syntax for making everything public:

```
{
      "public": ["/"],
}
```

Next, restart the metadata process with:

```
sudo supervisorctl restart metadata
```

Two domains should be seen on the quick start server's "Select Streams" when you access its IP address in a browser. The next section explains how to add user accounts and assign domain tags to them.

12 Add User Accounts/Assign Them “Domain” Tags

Add user accounts by running **`python accounts.py`**. The **`accounts.py`** file is located in the **`/mr-plotter/tools`** directory (**`cd mr-plotter/tools`** from **`/home/manager`** directory). The Mr. Plotter account access control console will open. Type **`help`** for additional clarification.

Usage:

```
adduser username password [tag1] [tag2] … [tag…n]
```

ex:

```
adduser peter psladmin psl lbnl pge
exit
```

The user “peter” has password “psladmin” and can access “psl”, “lbnl”, and “pge” domains. At least one tag is required when using **`adduser`** command.

Add additional “domain” tags to the end of the statement if user requires multiple access. All users by default can view what's tagged “public” in **tagconfig.json** (**`cd /home/manager/mr-plotter/tools`**).

The tags are the same domain tags that were updated in **upmuconfig.ini** (in the previous section), so “upmu”, "lbnl" or "psl" for instance. Next, edit **tagconfig.json** to add the tag that is assigned to the user and remove the access-controlled streams from public domain. Public can be empty set (no slash) if no MicroPMUs are to be made public.

The following example makes the **`/upmu`** domain public while only users with **`/lbnl`** domain access (**`adduser`** command) can view **`/lbnl`** MicroPMUs.

```
{
      "public": ["/upmu"],
      "lbnl": ["/lbnl"]
}
```

Run **`manager2lite.ipy`** and then restart services:

```
cd
pythonmanager2lite.ipy
```

(*adds metadata or updates existing plotter metadata*)

```
sudo supervisorctl restart all
```

Example:

1. Change the path domains for unit “uPMU_1” to **`/lbnl`** (from **`/upmu`**) in **`upmuconfig.ini`**
2. Run **`python accounts.py`** from **`/home/manager/mr-plotter/tools`** directory and **`adduser`** command.
3. Update **`tagconfig.json`** domains to reflect new “domain tags” and remove tags from public domain
4. Run **`python manager2lite.ipy`** and restart services with **`sudo supervisorctl restart all`**.
5. Domains now appear in "Select Streams" section. When logged in as a user with **`/lbnl`** domain permissions, the user will be able to view **`/lbnl`** domain streams. It is no longer publicly available as specified in **`tagconfig.json`**, unless streams are left public for demo purposes.

13 Clear Up Space by Deleting Synchronized Data

Delete files that have been synchronized to free up disk space using:

```
cd
./clean_synced.sh
```

Run this file, which will tell you how many files can be deleted. Type **`delete`** to verify data removal.

14 BTrDB Firewall (Shorewall) Reference

This guide covers step-by-step procedures for securing an Ubuntu 16.04 server running BTrDB using Shorewall following the BTrDB Firewall Reference. Shorewall comes installed on PSL Quick Start Kit, so installation instructions are for reference.

Double check that Shorewall is installed with the following command.

```
sudo shorewall status
```

If there is a line text in the returned message that shows “Shorewall is running”, then the installation section may be skipped.

14.1 Shorewall Installation

Please ensure that the system software is up-to-date as security vulnerabilities to exposed services cannot be prevented by a firewall.

```
sudo apt-get update

sudo apt-get dist-upgrade -y
```

Then install the shorewall package.

```
sudo apt-get install -y shorewall
```

Shorewall is disabled and unconfigured by default, copy the default configuration files and modify them.

```
cd /usr/share/doc/shorewall/examples/one-interface/

sudo cp rules interfaces zones policy /etc/shorewall/
```

14.2 Interfaces

The first step is to ensure that the interface described in **`/etc/shorewall/interfaces`** is correct. To see interfaces, run the following command.

```
ifconfig
```

The interface will have a name like **`eno1`** or **`eth0`**. Edit **`/etc/shorewall/interfaces`** so that the name matches. If the interface is configured statically, also remove the ‘dhcp’ from the interface flags. For reference, our file looks like this.

```
#ZONE      INTERFACE      OPTIONS

net        eno1           tcpflags
```

14.3 Rules

The default policy (deny all) and zones files are acceptable. It is necessary to modify the rules file to permit traffic to the BTrDB server. Add the following lines to **`/etc/shorewall/rules`**

```
# Allow access to the plotter

ACCEPT          net        $FW        tcp        80,443

# Allow uPMU data

ACCEPT          net        $FW        tcp        1883

# Allow access to the BTrDB API

ACCEPT          net        $FW        tcp        4410,9000
```

Note that the default shorewall rules file denies ICMP Pings to the server. You may want to remove this line.

```
Ping(DROP)      net        $FW
```

And replace it with

```
Ping(ACCEPT)    net        $FW
```

In addition, you may want to permit SSH access with

```
ACCEPT          net        $FW        tcp        22
```

14.4 Auto start

To ensure the firewall starts on boot, enable it in **`/etc/default/shorewall`**. Open that file and change the following line from

```
startup=0
```

To

```
startup=1
```

so that the firewall will start on boot.

Finally, add shorewall to systemctl with the following command.

```
sudo systemctl enable shorewall
```

Final Notes

If any key service updates were installed in the dist-upgrade step earlier, or a kernel update was installed, it is best to reboot the server so that these take effect.

```
sudo reboot
```

If there were no key updates, simply start the firewall without rebooting the server with the following command.

```
sudo shorewall safe-restart
```

15 Troubleshooting

Symptom:

No data is shown on the plotter.

Solution #1:

Data must be sent from MicroPMUs to **receiver**, which saves them to a queue for **sync2_quasar** services. Quasar is the database system. To check if **sync2_quasar** and **receiver** are running as expected, use the following commands.

```
sudo supervisorctl tail –f receiver
```

`Ctrl+c` (*exit tail program*)

```
sudo supervisorctl tail –f sync2_quasar
```

`Ctrl+c` (*exit tail program*)

Symptom:

Data is not being sent to the plotter.

Solution #1:

Check if receiver is running on the server. By default, it should constantly be receiving data from MicroPMUs.

```
sudo supervisorctl tail –f receiver
```

`Ctrl+C` (*exit tail program*)

Solution #2:

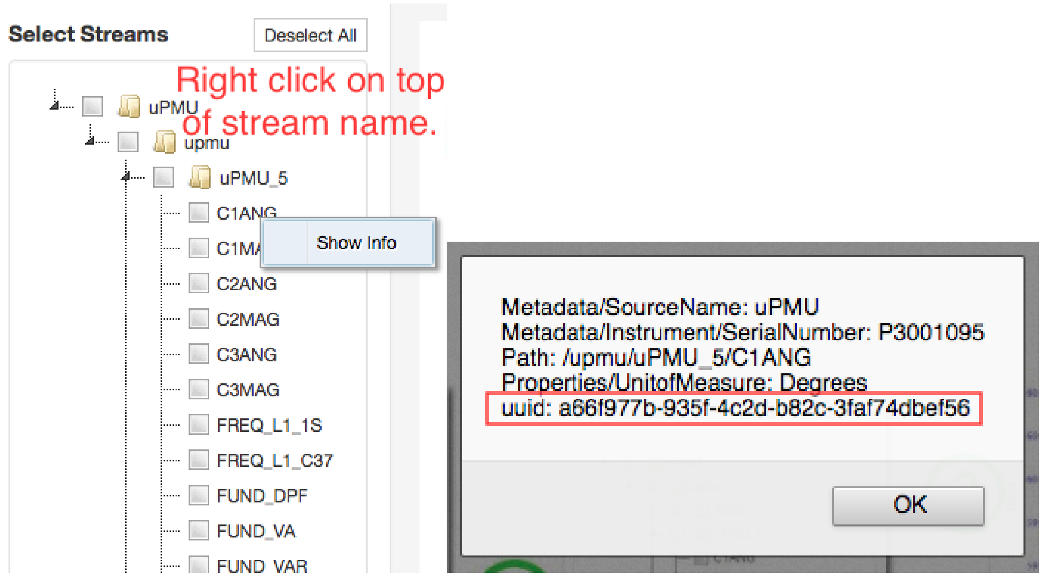
Transfer agent may have stopped running due to prolonged server outage. The transfer agent can be restarted by restarting your MicroPMU. Also make sure the MicroPMUs are sending data to the Quick Start server. Refer to "Routing MicroPMU data to the Quickstart Kit Server" for data routing instructions.

16 Appendices

16.1 BTrDB REST API Examples

BTrDB includes a REST api that can be used from any language supporting HTTP. This guide will explain how to do some common operations using curl which is a command line program, usually available by default on MacOSX (unless using a very old version of OSX), but can also be installed on Windows. Bear in mind that the binary API is faster, and has golang and python support.

BTrDB stores data in streams identified by a UUID (Universally Unique Identifier) which can be found on the plotter website by right clicking on a stream name and clicking "show info".



```
Metadata/SourceName: uPMU
Metadata/Instrument/SerialNumber: P3001095
Path: /upmu/uPMU_5/C1ANG
Properties/UnitofMeasure: Degrees
uuid: a66f977b-935f-4c2d-b82c-3faf74dbef56
```

16.1.1 Optional jq Installation

The BTrDB REST API returns data in JSON (javascript object notation) format. While not necessary, it is recommended to install jq which will parse the JSON data into a more readable format when API requests are made. The jq binaries can be downloaded from https://stedolan.github.io/jq/ for Windows, Linux or Mac OSX.

For Windows, simply download the .exe file and save it to the C: drive and add the path to the .exe to the system's "PATH" environment variable. Refer to the curl installation in the next section for instructions on editing the system environment variables.

For Mac OSX, download the appropriate binary file and move it into your **`/usr/bin/`** directory with the **`mv`** command. Note that execute permissions may need to be added to the binary with the following command.

```
        sudo chmod +x /path-to-jq-binary
```

If any permission errors are raised, prepend the commands with **`sudo`** and enter the administrator password (password used to log on to the Mac).

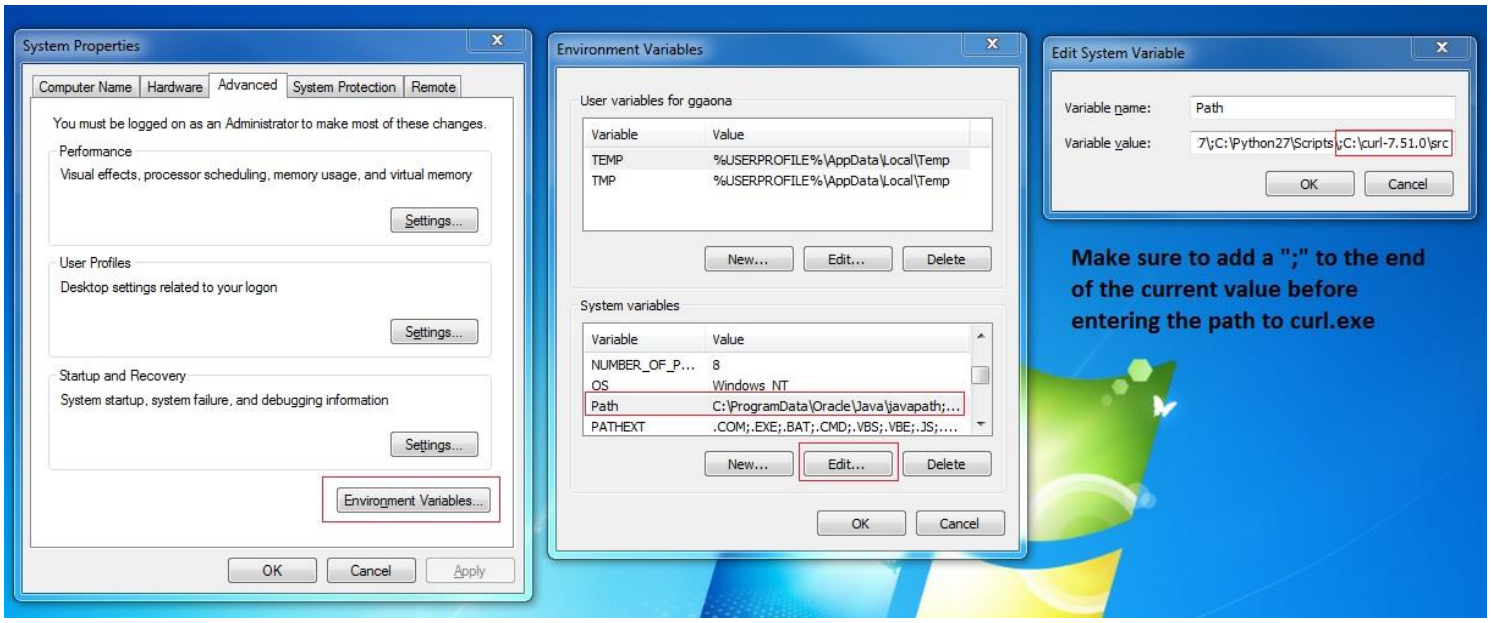
16.1.2 Using Curl

16.1.2.1 Windows Install

Curl can be installed on Windows by downloading the source from http://curl.haxx.se/download.html. Get either the generic win32 or win64 package depending on whether the Windows OS being used is a 32-bit or 64-bit OS. Note that the latest version(s) should be highlighted in yellow.

1. Create a "curl" folder in the "C:" drive and extract the contents of the source zip file to that folder.
2. Now open the start menu and in the search bar type "edit the system environment variables" and press the enter key.
3. In the window that appears click on the "Environment Variables" button toward the bottom right corner.
4. In the next window find the "Path" variable and click the "Edit" button.
5. Finally add the path of the curl.exe to the end of the variable value which should be C:\curl\src\ if the source zip was extracted into a curl folder on the C drive.

See the following image for clarification.



Variable	Value
TEMP	%USERPROFILE%\AppData\Local\Temp
TMP	%USERPROFILE%\AppData\Local\Temp

Variable	Value
NUMBER_OF_P...	8
OS	Windows_NT
Path	C:\ProgramData\Oracle\Java\javapath;...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;....

Variable name: Path

Variable value: 7\;C:\Python27\Scripts;C:\curl-7.51.0\src

Make sure to add a ";" to the end of the current value before entering the path to curl.exe

Now curl can be run from the Windows command line. Note that variables can be set and accessed in the Windows command line like so.

```
set foo=bar
```

(set value of "bar" to variable "foo")

```
echo %foo%
```

(prints value of variable "foo")

```
set "foo="
```

(removes value for variable "foo")

16.1.2.2 API Examples

16.1.2.2.1 Checking Status

To verify that the server is up, and the firewall is correctly configured, use the status endpoint.

```
SERVER=http://127.0.0.1:9000
```

(replace 127.0.0.1 with the Quick Start server IP)

```
curl $SERVER/status
```

(prints OK to the command line if the server is not experiencing any issues)

If an error message is returned, first double check that the correct IP is being used. Otherwise, the server is either on a different port or the firewall is incorrectly configured. At that point, please contact support for further assistance.

16.1.2.2.2 Querying Data

With a stream UUID, either a raw query or statistical query may be performed. Raw queries retrieve data at its native resolution and statistical queries which obtains coarser resolution but is substantially faster.

To get an idea of how much data is available, a bracket query may be used and is formatted like so.

```
curl –X POST $SERVER/q/bracket –d '{"UUIDS": ["UUID-Goes-Inside-
Quotes"]}'
```

Example:

```
SERVER=http://127.0.0.1:9000
```
(replace with Quick Start server IP)

```
UUID=f0896731-0e43-4ed5-a97f-19266d8e732c

curl –X POST $SERVER/q/bracket –d '{"UUIDS": ["'$UUID'"]}' | jq
```

(`jq` command is optional and used to print returned data in more readable format. See the "Optional jq Installation" section for more info. Also note that the $UUID variable must be wrapped in "'" double and single quotes)

Windows Example (minor differences in how variables are accessed and escaped characters):

```
set SERVER=172.17.6.122:9000

set UUID=f0896731-0e43-4ed5-a97f-19266d8e732c

curl –X POST %SERVER%/q/bracket –d "{\"UUIDS": [\"%UUID%\"]}" | jq
```

The server will return data in JSON format if it recognizes the request as a valid request.

```
{
      "Brackets": [
            [
                  1477100329040881700,
                  1477186729034743600
            ]
      ],
      "Merged": [
            1477100329040881700,
                    1477186729034743600
                ]
            }
```

The data in brackets is the start and end time of the stream's current recording period in Unix time (measured in nanoseconds). The data in merged is the start and end time of all queried streams' current recording time period.

A point width parameter can also be used in conjunction with the start and end time of the stream to perform a statistical query. The point width sets the resolution of the data returned, meaning that it will return statistics in the interval set by the point width value. Point width values are equal to log base 2 of a time interval in nanoseconds. For example, pw=23 is 2^23 nanoseconds or 8.388 ms. Below is a table of point width reference values.

PW	Rough Equivalent	Actual Value
23	8.388 ms	16,777,216
30	1 second	1,073,741,824
36	1 minute	68,719,476,736
42	1 hour (73 minutes)	4,398,046,511,104
46	1 day (19.5 hours)	70,368,744,177,664
49	1 week (6.5 days)	562,949,953,421,312
51	1 month (26 days)	2,251,799,813,685,248
54	1 year (208 days)	18,014,398,509,481,984

BTrDB does support exact windowing, although not through the REST API. Point widths are useful because they allow for significantly faster query responses than exact windowing through the binary interface.

A statistical query of the example stream will be structured as follows.

```
STIME=1477100329040881700
ETIME=1477186729034743600
```

`PW=46` *(returns data in 1 day intervals)*

```
PARAMS=”starttime=$STIME&endtime=$ETIME&unitoftime=ns&pw=$PW”
UUID=f0896731-0e43-4ed5-a97f-19266dd8e732c
```

`curl –X GET “$SERVER/data/uuid/$UUID?$PARAMS” | jq` *(pipe to jq command is optional)*

Windows Example:

```
set STIME=1477100329040881700
set ETIME=1477186729034743600
```

`set PW=46` *(returns data in 1 day intervals)*

```
set PARAMS=”starttime=%STIME%&endtime=%ETIME%&unitoftime=ns&pw=%PW%”
set UUID=f0896731-0e43-4ed5-a97f-19266dd8e732c
```

`curl –X GET %SERVER%/data/uuid/%UUID%?%PARAMS% | jq` *(pipe to jq command is optional. Also note the path is not encased in quotes for the Windows version of this command)*

The response will look like this.

```
[
      {
            “uuid”: “f0896731-0e43-4ed5-a97f-19266dd8e732c”,
            “XReadings”: [
                  [
                        1477039940289,
                        167360,
                        95915605781875420,
                        95915929807604350,
                        95916253833333300,
                        1197648
                  ],
                  [
                        1477110309033,
                        345024,
                        95916253833874380,
                        95918538533067200,
                        95920823232259970,
                        8444587
                  ]
```

```
                ],
                "version": 2
            }
        ]
```

Since there are only two recordings, this indicates the total recording period for the stream is roughly two days since our point width interval was about one day. Each recording contains the following six fields.

- Timestamp in nanoseconds since the epoch
- The remainder of the timestamp in nanoseconds
- The minimum recorded value of the stream
- The average recorded value of the stream
- The maximum recorded value of the stream
- The number of values

The **jq** command can be used to extract parts of this data. For example, retrieve the number of values with the same command, then pipe to **jq** like so.

```
        curl … | jq .[0].XReadings[][5]
```

Which will return only the sixth field for every reading (in this case, the number of values).

Statistical queries can also return results in CSV format. The endpoint for the CSV response accepts JSON parameters rather than URL parameters. Below is an example.

```
        SERVER=http://127.0.0.1:9000  (replace with Quick Start server IP)
        curl –X POST $SERVER/directcsv –d
        '{
            "UUIDS":["'$UUID'"],        (make sure to wrap in "'" double quote, single quote)
            "Labels": ["C1ANG"],
            "StartTime": '"$STIME"',  (make sure to wrap in '"' single quote, double quote)
            "EndTime": '"$ETIME"',
            "UnitofTime": "ns",
            "PointWidth": 46
        }' > file.csv  (where file.csv is the name of the output file to write to)
```

Widows Example (Note the command must be on one line in Windows, although the example is split in multiple lines for readability):

```
curl –X POST %SERVER%/directcsv –d
"{
      \"UUIDS\":[\"%UUID%\"],
      \"Labels\": [\"C1ANG\"],
      \"StartTime\": \"%STIME%\",
      \"EndTime\": \"%ETIME%\",
      \"UnitofTime\": \"ns\",
      \"PointWidth\": 46
}" > file.csv
```
(where file.csv is the name of the output file to write to)

Finally, over small periods of time, raw data can be queried. This looks the same as the statistical query, but does not include the point width parameter. Below is an example.

```
STIME=1474321947545300000  (9/19/16 2:52:28 PM)
ETIME=1474326345591810000  (9/19/16 4:05:46 PM)
PARAMS="starttime=$STIME&endtime=$ETIME&unitoftime=ns"
UUID=f0896731-0e43-4ed5-a97f-19266d8e732c
curl –X GET "$SERVER/data/uuid/$UUID?$PARAMS" > data.json
```

Note that even for this small time period (~1 hour), the file is quite large (10MB). Take care not to perform raw queries over too much data. Example of file information output via `ls` command.

```
$ ls –hal data.json
$ -rw-rw-r-- 1 userA userA 10M Oct 23 13:05 data.json
```

Windows Example:

```
set STIME=1474321947545300000  (9/19/16 2:52:28 PM)
set ETIME=1474326345591810000  (9/19/16 4:05:46 PM)
set PARAMS="starttime=%STIME%&endtime=%ETIME%&unitoftime=ns"
set UUID=f0896731-0e43-4ed5-a97f-19266dd8e732c
curl –X GET %SERVER%/data/uuid/%UUID%?%PARAMS% > data.json
```

16.2 BTrDB Binary API Examples

16.2.1 Binary API via Python

NOTE: It is assumed that users of the python interface to the binary API are familiar with the python programming language. Otherwise it is recommended to use the REST API instead.

The python interface for the binary API is supported on Linux and Mac OSX, but not Windows. Before installing the `btrdbcapnp` module, make sure `pip` (python package manager) is installed. Installation instructions for `pip` can be found at https://pip.pypa.io/en/stable/installing/.

To install the `btrdbcapnp` module, first download the module from https://github.com/SoftwareDefinedBuildings/btrdb-python/archive/master.zip. Once downloaded, extract the contents of the zip file then move into the extracted folder from the command line and run the setup.py file.

```
cd path-to-extracted-zip/btrdb-python-master/
```

```
python setup.py install
```

Once installed, the binary API can be queried with python. Check the readme file for an example of using the `btrdbcapnp` module. You can also view documentation of the module with the following command.

```
pydoc btrdbcapnp
```

16.2.2 Binary API via Go

NOTE: It is assumed that users of the Go interface to the binary API are familiar with the Go programming language. Otherwise it is recommended to use the REST API instead.

The Go interface to the binary API can be installed with the following command.

```
go get github.com/go-btrdb/btrdb
```

An example program for interacting with the binary API can be found in the source of `btrdb.go`.

For technical assistance or the latest version of this document, please call 1-510-522-4455 or email us at: support@powersensorsltd.com if you have any further questions.