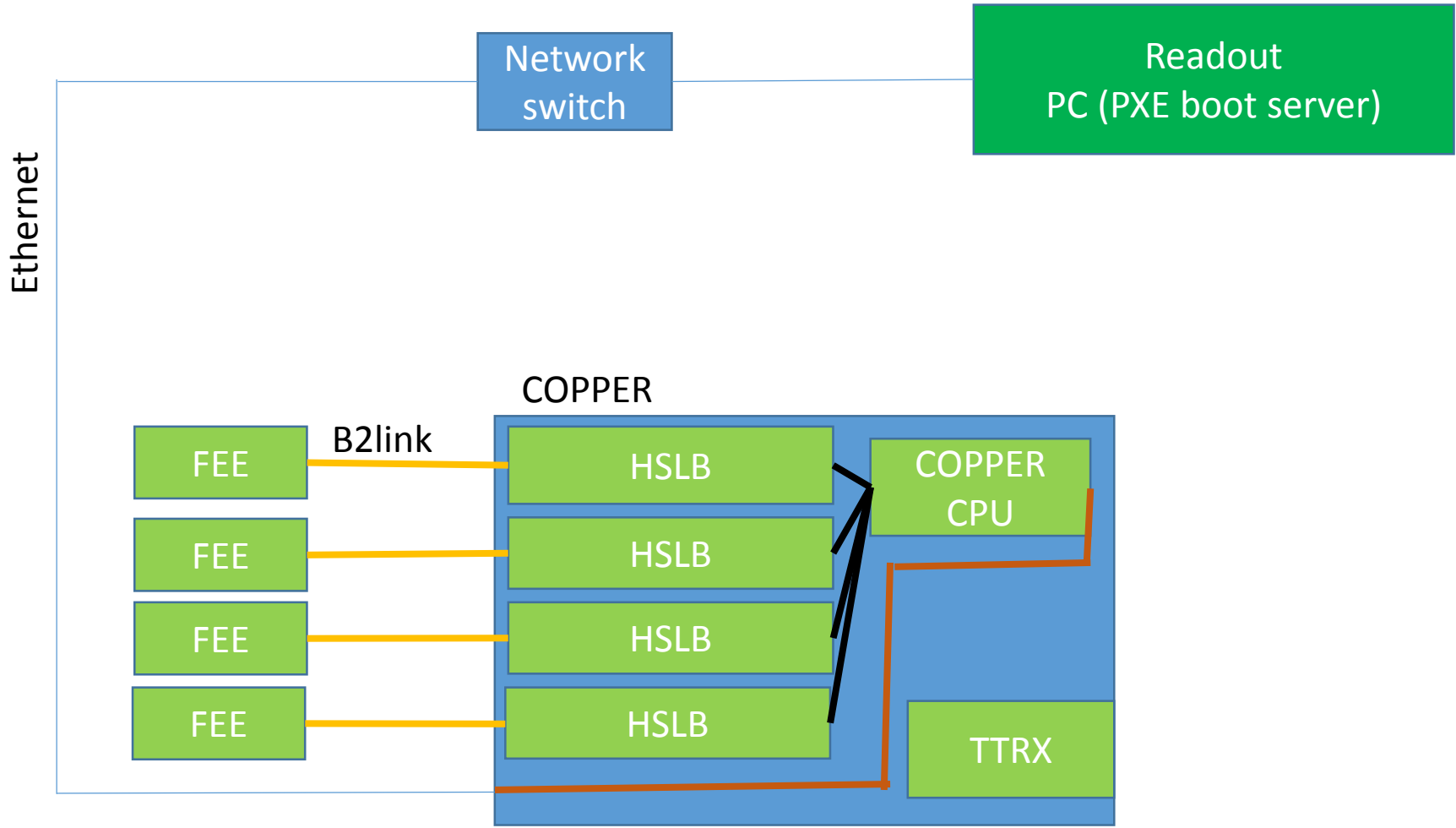


# Pocket DAQ manual (rev.13542)

Oct.29, 2014

Satoru Yamada

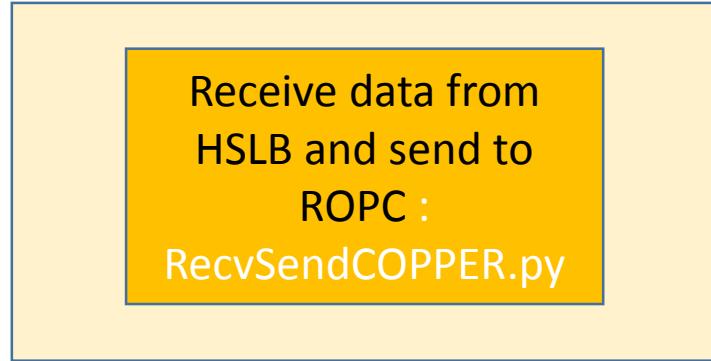
# Connection between components



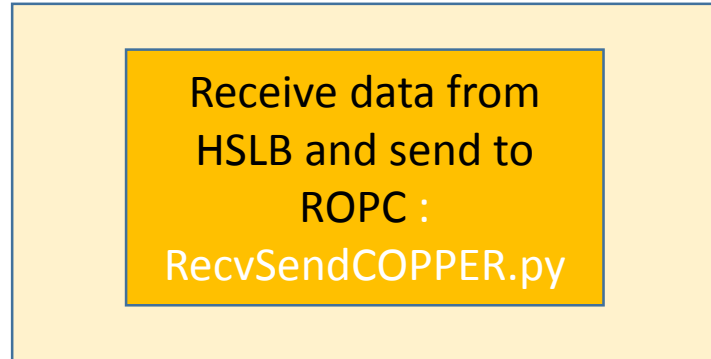
# Old Software components

These are stored under <https://belle2.cc.kek.jp/svn/trunk/software/daq>

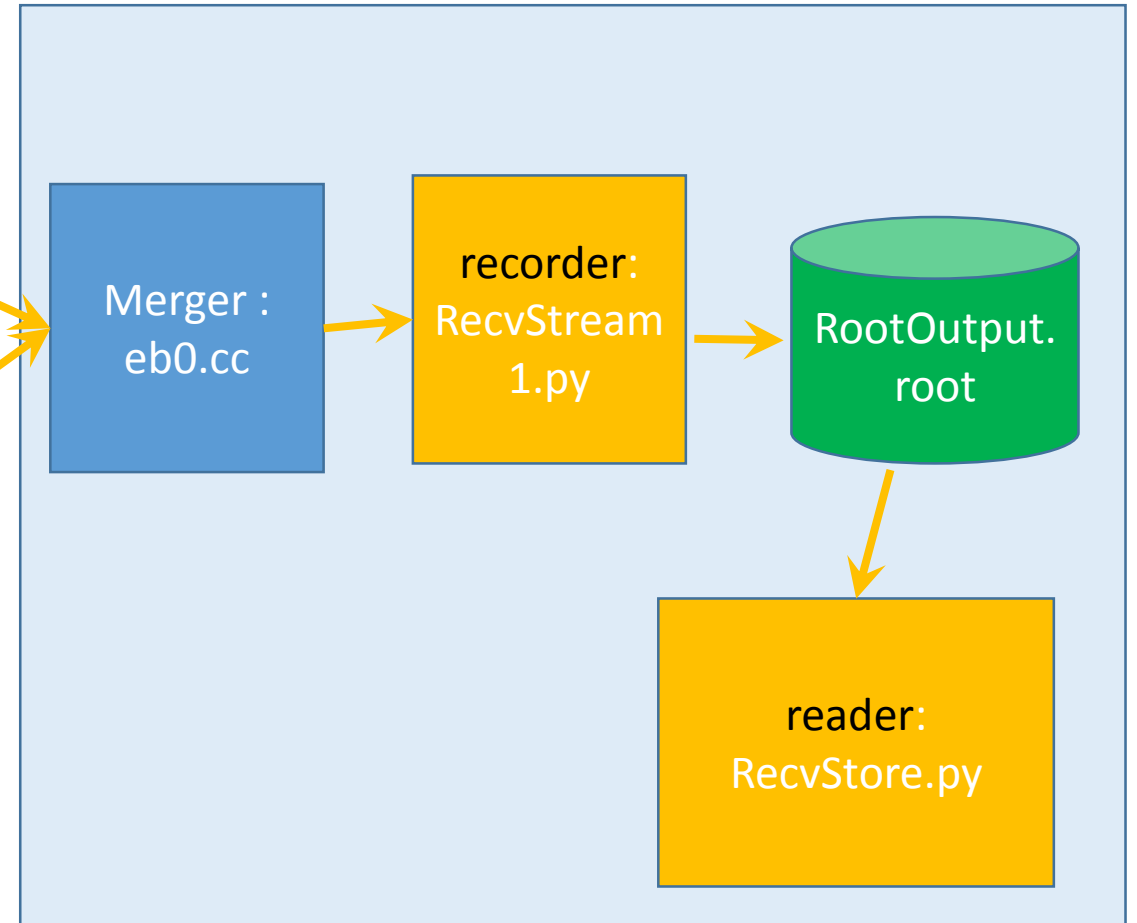
COPPER CPU



COPPER CPU



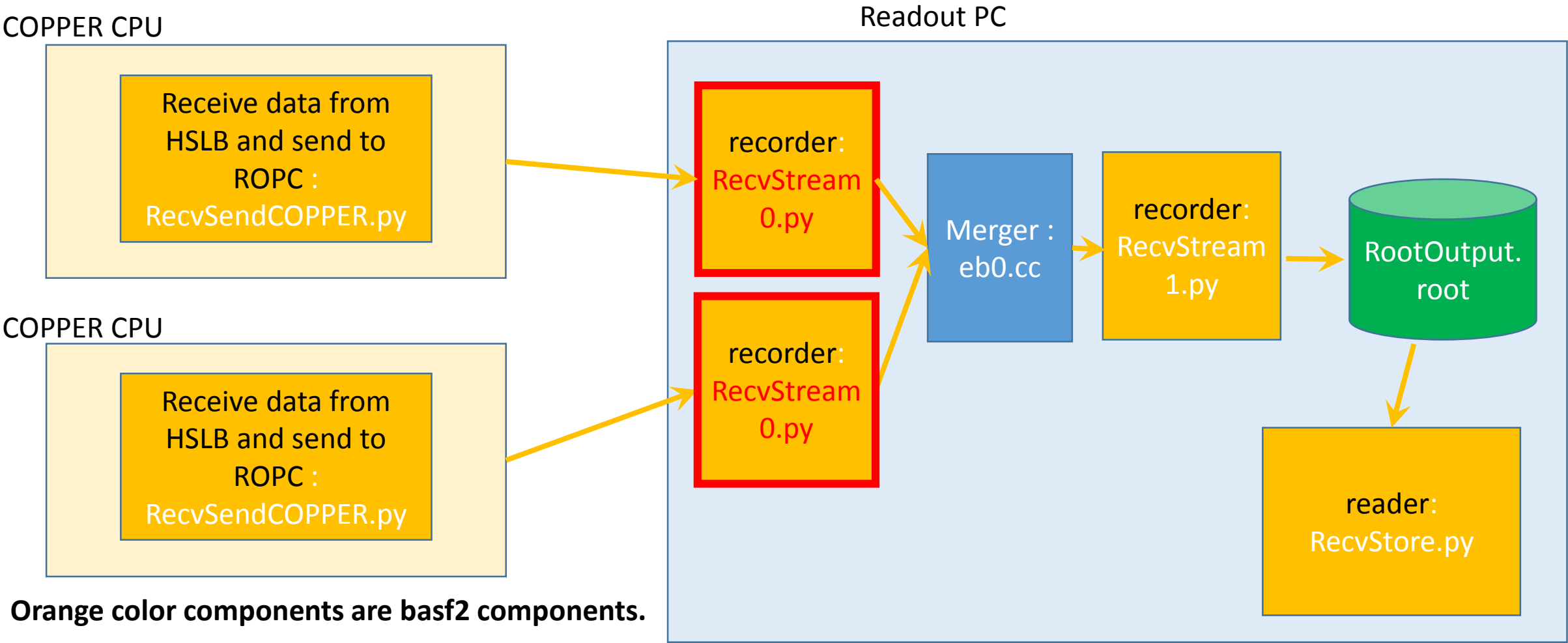
Readout PC



Orange color components are basf2 components.

# New Software components (from rev. 13542)

These are stored under <https://belle2.cc.kek.jp/svn/trunk/software/daq>



Pocket DAQ w/o slow  
controller and GUI

# 0. Before using pocket DAQ

- Setup a PXE boot server for COPPER CPU and install driver for COPPER etc
  - See <https://belle2.cc.kek.jp/~twiki/bin/view/Detector/DAQ/PocketDAQ>
- Install basf2 on both COPPER CPU and Readout PC
  - [See https://belle2.cc.kek.jp/~twiki/bin/view/Software/SoftwareInstallation](https://belle2.cc.kek.jp/~twiki/bin/view/Software/SoftwareInstallation) .  
“addall” does not install the daq package. You need to do “addpkg daq”.
  - Check daq/Sconscripts and check if `env['CONTINUE'] = False` is commented out.
  - Compile with scon.
- Compile eventbuilder
  - `cd ${BELLE2_LOCAL_DIR}/daq/eventbuilder/evb0/ ; gmake eb0`

# Install GLEW for running basf2 program

After external ver.0.5.0?, basf2 requires GLEW(The OpenGL Extension Wrangler Library).

The following procedure worked fine.

1, install the yum epel repository.

```
ropc01$ wget http://ftp-srv2.kddilabs.jp/Linux/distributions/fedora/epel/5/i386/epel-release-5-4.noarch.rpm
```

```
ropc01$ ssh cpr** -lroot
```

```
cpr**# rpm -ivh epel-release-5-4.noarch.rpm
```

```
cpr**# ls -l /etc/yum.repos.d/
```

```
cpr**# exit
```

2, install glew on COPPER

```
ropc01$ sudo yum install glew --installroot=/tftpboot/copper/root
```

# 1, Set parameters in run\_start.sh (1)

```
[ROPC] % cd ${BELLE2_LOCAL_DIR}/daq/copper/daq_scripts
```

```
[ROPC] % emacs run_start.sh
```

## Set arguments of start\_copper.sh

Usage : start\_copper.sh **<HOSTNAME>** **<COPPER node>** **<FINNESSE bit flag: A=1, B=2, C=4, D=8>** **<Use network shared memory : Yes=1, No=0>** **<node name(used for Network Shared Memory)>**

e.g.)

```
/usr/bin/xterm -e ${BELLE2_LOCAL_DIR}/daq/copper/daq_scripts/start_copper.sh cpr006 1 1 0 0&
```

```
/usr/bin/xterm -e ${BELLE2_LOCAL_DIR}/daq/copper/daq_scripts/start_copper.sh cpr007 2 3 0 0&
```

<COPPER node ID> will be attached to RawCOPPER header.



# 1, Set parameters in run\_start.sh (2)

## **NOTICE : Environment variables for basf2**

In `${BELLE2_LOCAL_DIR}/daq/copper/daq_scripts/copper.sh`, a line, “source ~/.bash\_profile”, is for setting up basf2 environment. You need to add basf2 setting commands in your `.bash_profile` (or other script file).

**Please see "Setup of Software Tools" at <https://belle2.cc.kek.jp/~twiki/bin/view/Software/SoftwareInstallation> for details.**

```
[ ${BELLE2_LOCAL_DIR}/daq/copper/daq_scripts/copper.sh ]
```

```
#
```

```
# setup basf2 environment (See "Setup of Software Tools" at
```

```
https://belle2.cc.kek.jp/~twiki/bin/view/Computing/SoftwareInstallation
```

```
#
```

```
source ~/.bash_profile
```

# 1, Set parameters in run\_start.sh (3)

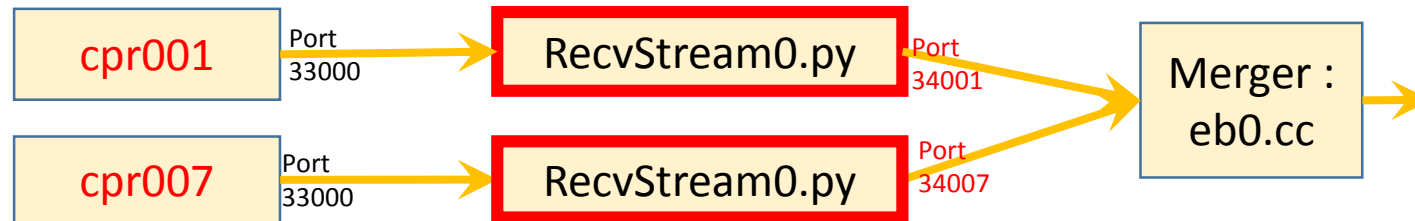
## Set arguments of RecvStream0.sh

Usage : RecvStream0.py <COPPER hostname> <Use NSM(Network Shared Memory)? yes=1/no=0> <port # to accept connection from eb0> <NSM nodename>

Example :

/usr/bin/xterm -e "\${BELLE2\_LOCAL\_DIR}/daq/copper/daq\_scripts/RecvStream0.sh cpr001 0 34001 hogehoge1000;" &  
/usr/bin/xterm -e "\${BELLE2\_LOCAL\_DIR}/daq/copper/daq\_scripts/RecvStream0.sh cpr007 0 34007 hogehoge1000;" &

Readout PC



# 1, Set parameters in run\_start.sh (4)

**Specify whether communication with GUI via shared memory will be used or not :**

## **1, Edit \${BELLE2\_LOCAL\_DIR}/daq/rawdata/examples/RecvStream1.py**

If you use GUI, set 'UseShmFlag' = 1

```
receiver.param('UseShmFlag', 1)
```

If you don't use GUI set 'UseShmFlag' = 0

```
receiver.param('UseShmFlag', 0)
```

## **2, Edit \${BELLE2\_LOCAL\_DIR}/daq/rawdata/examples/RecvStream0.py**

If you use GUI, set 'UseShmFlag' = 1

```
receiver.param('UseShmFlag', 1)
```

If you don't use GUI set 'UseShmFlag' = 0

```
receiver.param('UseShmFlag', 0)
```

## 2, Setting of merger (eb0) (1)

### 1, compile

```
$ cd ${BELLE2_LOCAL_DIR}/daq/eventbuilder/evb0
```

```
$ make
```

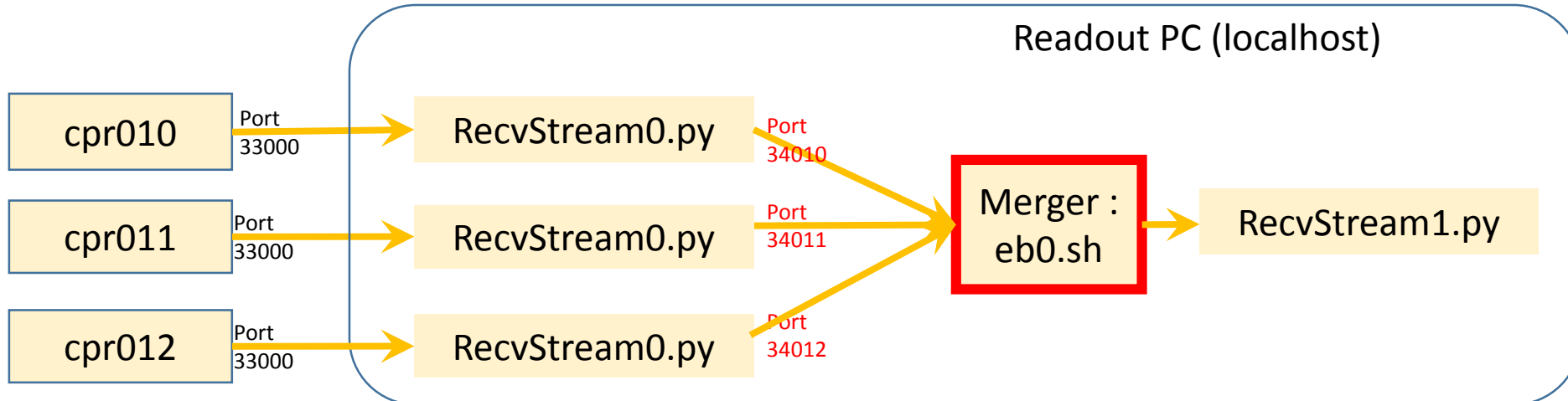
### 2, edit eb0.sh

`cd /home/usr/b2daq/eb` # change to `cd "your local BelleII directroy"/daq/eventbuilder/evb0/eb-xinetd`  
( e.g. `cd /home/usr/yamada/basf2/release/daq/eventbuilder/evb0/eb-xinetd` )

If you want to read data from COPPERs with hostnames of cpr010, cpr011 and cpr012 and corresponding port # of RecvStream0.py are 34010, 34011 and 34012, respectively, use the following ;

~~`./eb0 cpr010:33000 cpr011:33000 cpr012:33000`~~

`./eb0 localhost:34010 localhost:34011 localhost:34012`



## 2, Setting of merger (eb0) (2)

**3, edit eb0's part in \${BELLE2\_LOCAL\_DIR}/daq/eventbuilder/evb0/eb-xinted**

Modify **user** and **server**.

service eb0

```
{
    bind = 127.0.0.1
    socket_type = stream
    type = UNLISTED
    port = 5101
    wait = no
    user = g0cdc # Change to your user name (e.g. yamagata)
    server = /home/usr/b2daq/eb/eb0.sh
    # change to "your Belle2 local directory"/daq/eventbuilder/evb0/eb0.sh
    # (e.g. /home/usr/yamagata/basf2/release/daq/eventbuilder/evb0/eb0.sh)
}
```

You need to change **the other services in eb-xinetd** ( eb1tx, eb1rx, eb2tx, eb2rx .... ) **in the same way.**

## 2, Setting of merger (eb0) (3)

### **4, Set link to eb-xinetd**

```
# cd /etc/xinetd.d/  
# ln -s ${BELLE2_LOCAL_DIR}/daq/eventbuilder/evb0/eb-  
xinetd eb-xinetd  
# /sbin/service xinetd restart
```

The eb0 daemon will be automatically invoked when a basf2 program on a readout PC connects to a port specified in eb-xinted.

### 3, How to start DAQ

```
[ROPC] % cd ${BELLE2_LOCAL_DIR}/release/daq/copper/daq_scripts
```

```
[ROPC] % ./run_start.sh
```

## 4, How to stop DAQ

- No stop button for now (If you use GUI, you have.)
- You need to specify max # of events or time to stop the run on a basf2 python file.

```
ROPC % cd ${BELLE2_LOCAL_DIR}/daq/rawdata/examples  
Edit RecvStream1.py
```

You can set following paramters to stop a run.

```
receiver.param('MaxTime', 300.)
```

```
receiver.param('MaxEventNum', 30.)
```

-1 means infinite.



## 5. Read an output file and extract FEE buffer

- Output file name
  - `${BELLE2_LOCAL_DIR}/daq/copper/daq_scripts/root_output.root`
  - You can change the filename by editing `${BELLE2_LOCAL_DIR}/daq/rawdata/examples/RecvStream1.py`
- Read the root file
  - Please see dataformat and unpacker manuals.
  - Data format
    - [https://belle2.cc.kek.jp/svn/trunk/software/daq/copper/doc/SetupPocketDAQ\\_4p1\\_RawCOPPERDataHandling.pdf](https://belle2.cc.kek.jp/svn/trunk/software/daq/copper/doc/SetupPocketDAQ_4p1_RawCOPPERDataHandling.pdf)
  - Unpacker
    - [https://belle2.cc.kek.jp/svn/trunk/software/daq/copper/doc/SetupPocketDAQ\\_4p5\\_RawDataUnpackerPacker.pdf](https://belle2.cc.kek.jp/svn/trunk/software/daq/copper/doc/SetupPocketDAQ_4p5_RawDataUnpackerPacker.pdf)

# Revision history

- 2013/10/16
  - Change explanation about option of run\_start.sh script
- 2014/7/15 (rev. 11619)
  - To specify FINESSE bit field, you need to use decimal value ( not hexadecimal ).
  - Add an argument of RecvSenDCOPPER.py
  - Update eb0 related issues.
    - About eb0.sh
    - About eb0-xinted
- 2014/7/16(rev. 11654)
  - eb0.sh and eb-xinetd need more modification
- 2014/7/19
  - Add a comment about “addpkg daq”
- 2014/8/8
  - Modify usage of start\_run.sh. (nodename is only used for NSM)
- 2014/10/29
  - Change a DAQ scheme so that RecvStream0.py is used on a readout PC

# Appendix

# A-1, yum setting for COPPER

Sometime yum fails when you try to install software to COPPER directory;

```
readoutPC # yum install **** --installroot=/tftpboot/copper/root
```

In that case, you need to change yum setting for COPPER host:

1, Copy yum repository files.

```
readoutPC # cp /etc/yum.repo.d/* /tftpboot/copper/root/etc/yum.repo.d/*
```

2, Set a proxy server in /tftpboot/copper/root/etc/yum.conf **same as** readoutPC:/etc/yum.conf

## A-2, Socket parameters on COPPER, readout PC and etc..

1, Enable "Jumbo Frame" in network switch's setting, if necessary (e.g. HP1810-24G)

If the switch does not support Jumbo frame, setting MTU a large value may cause a problem.

2, Change socket parameters

```
# /sbin/ifconfig eth? mtu 9000
```

```
# /sbin/sysctl -w net.ipv4.tcp_rmem="8388608 8388608 8388608"
```

```
# /sbin/sysctl -w net.ipv4.tcp_wmem="8388608 8388608 8388608"
```

```
# /sbin/sysctl -w net.ipv4.tcp_mem="9000000 9000000 9000000"
```

```
# /sbin/sysctl -w net.core.rmem_default=8388608
```

```
# /sbin/sysctl -w net.core.wmem_default=8388608
```

```
# /sbin/sysctl -w net.core.rmem_max=8388608
```

```
# /sbin/sysctl -w net.core.wmem_max=8388608
```

3, edit /etc/rc.local(or /tftpboot/copper/root/etc/rc.local ) if you want the parameters automatically set at boot time

## A-3, Use ntpd to adjust time on COPPER

1, Edit /tftpboot/copper/root/etc/rc.d/init.d/ntpd

```
# Source networking configuration.
```

```
. /etc/sysconfig/network
```

```
ntpdate 192.168.10.1
```

```
if [ -f /etc/sysconfig/ntpd ];then
```

```
    . /etc/sysconfig/ntpd
```

```
fi
```

2, check ntpd is started at boot time

```
cpr*** $ chkconfig --list | grep ntp
```

## A-4, syslogd

Install syslogd :

readoutPC # yum install syslogd --installroot=/tftpboot/copper/root

Add entry to /tftpboot/copper/root/etc/syslogd.conf if needed:

# Log anything (except mail) of level info or higher.

# Don't log private authentication messages!

\*.info;mail.none;news.none;authpriv.none;cron.none      /var/log/messages

## B-1, Test bench at Tsukuba B3

