# MuCap Berkeley Analysis Software
# User Manual

Tom Banks (*tbanks@socrates.berkeley.edu*)
Fred Gray (*fegray@socrates.berkeley.edu*)
University of California, Berkeley

August 31, 2003

## 1 Introduction

The MuCap Berkeley Analysis Software (MBAS) is a MIDAS-based analysis package. More specifically, the MBAS makes use of the MIDAS Analyzer framework. For detailed documentation on the MIDAS Analyzer, we refer the user to the website

http://midas.triumf.ca/doc/analyzertask.html

For our purposes in this manual, a brief description will suffice:

The $\mu$Cap raw data blocks are subdivided into *banks*, named according to the electronics module from which the raw data was drawn (TDC400s, CAENs, etc.). The MIDAS Analyzer makes use of the "multi-stage" concept when processing a data block (Figure 1). At each stage one of the MBAS *modules* acts upon one or more data banks, usually performing some specific task. The results from the module's calculations are placed into a new bank, which is added to the list of existing banks. The first stages typically process and prepare the raw data for subsequent histogram-filling stages. The multi-stage system is especially useful because it allows for standard collaboration preprocessing stages, while later analysis stages can be customized by individuals. It is possible to write analysis modules in either C++ or Fred's MQL (Muon Query Language).

## 2 Installing MIDAS, ROOT, & the Java Development Kit

If you intend to run the MBAS on the PSI LLC or Merlin clusters, this step has already been done for you. The supporting software has been installed on the PSI `afs` cluster at:

`/afs/psi.ch/project/mucap/`

If you intend to run the MBAS elsewhere, you will need to install and configure all of the supporting software yourself. If you need help in doing this, contact Fred for assistance and advice.

## 3 Installing the MBAS

There are two steps necessary to set up the MBAS on your computer:

- **Step 1 – Configure your `.bash_profile` and `.bashrc` files**

  We have composed a `.bash_profile` template which should allow the MBAS to function on the PSI LLC computing cluster (PSI Red Hat 6.2), the PSI Merlin cluster (PSI Red Hat 6.2, plus extra stuff), or any machine running standard PSI Red Hat 7.3. Replace or merge your existing `.bash_profile` with this template. This will configure your account to use the correct versions of `gcc` (i.e. the C, C++, and Fortran compilers), ROOT, and MIDAS.

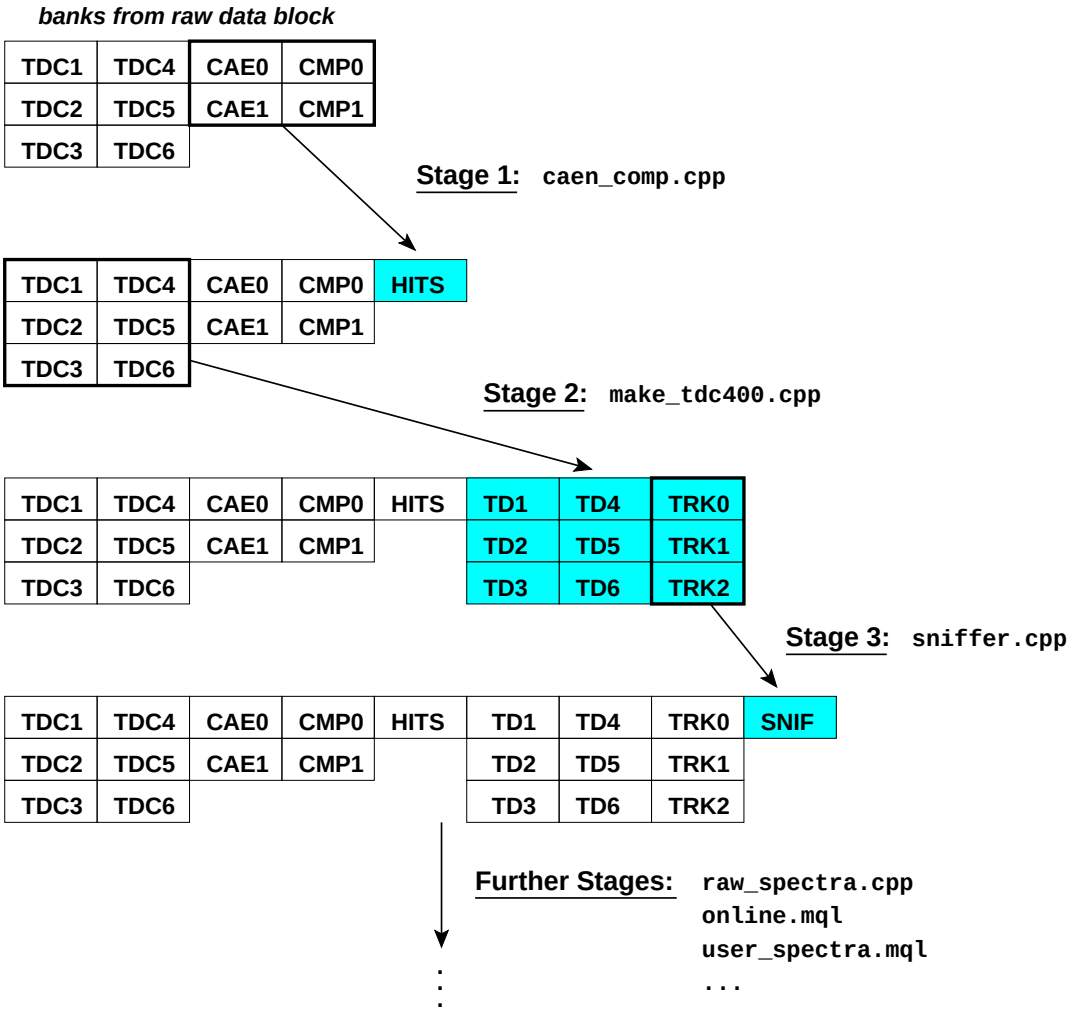


Figure 1: Schematic of MIDAS Analyzer multi-stage action on $\mu$Cap data.

In addition, inspect your `.bashrc` file, and make sure that the specified CERN library path is accurate.

Remember that you must log out and back in again in order for these changes to take effect.

- **Step 2 – Establish the CVS directory**

  Now you are ready to check out a copy of the current MBAS from the CVS repository. The command

  ```
  cvs -d mucap@kaon.physics.berkeley.edu:/home/mucap/cvsroot co mu
  ```

  will create a directory named `mu/` and fill it with a copy of the current MBAS. (*Note*: The above command will prompt you for a password; enter the common $\mu$Cap `data` password.)

# 4 Running the MBAS – Quickstart

Once the MBAS directory `mu/` has been established, move into it and compile the MBAS software (*Note*: Do not type the "$>$" character! It refers to the Linux prompt):

```
> cd mu
mu> make
```

This will create an executable analysis file in the subdirectory `work`. The executable file is (unfortunately) also named "`mu`". Before we can run the executable, we must prepare the MIDAS ODB by creating an `Equipment` directory:

```
mu> odbedit
[ODBedit]> mkdir Equipment
[ODBedit]> exit
```

Now we are ready to begin running the MBAS. Move into the `work` subdirectory, and run the executable "`mu`" on an appropriate file:

```
mu> cd work
work> ./mu -i /archiv/user/p/petitjean/r2003a/run00999.mid -o 999.root -n 20
```
[1, 2]

In the command above, the location of the input file is given by the "`-i`" flag, and the location and name of the output file is specified by the "`-o`" flag. The "`-n`" flag is often useful: it allows one to analyze only the first *n* blocks of a run. To see a comprehensive list of the available options, type

```
work> ./mu -help
```

[1] From our experience, it is necessary to run the executable two or three times before it works properly–it takes several attempts for the MIDAS analyzer to "learn" what it needs to do. Thus, after a failed attempt you will need to cancel the stalled process with CTRL-C and perform the command "`clean`" in `odbedit`, before retrying.

[2] Once the analysis is running, you may observe some output messages like the following:

```
[mana.c:3389:write_event_ttree] Received unknown bank TDC1
[mana.c:3389:write_event_ttree] Received unknown bank TDC2
[mana.c:3389:write_event_ttree] Received unknown bank TDC3
                    ...
```

This is really just a cosmetic problem, which can be eliminated by going into `odbedit` and typing the following:

```
[ODBedit]> cd Analyzer
[ODBedit]> set ''Book TTree'' n
[ODBedit]> cd Output
[ODBedit]> set ''Events to ODB'' n
[ODBedit]> quit
```

The output file *output_name*.root can be viewed with standard ROOT software. ROOT usage will not be discussed in this document; please refer to the ROOT User's Manual for any questions.

# 5 Using the MBAS – Detailed

This section is intended to provide an explanation of exactly how the MIDAS Analyzer and MBAS function. If you have not already worked through the previous Installation and Quickstart sections of this document, it is recommended that you now do so.

## 5.1 The MBAS Structure

The MBAS directory mu/ (created in Step 2 of section "Installing the MBAS"), contains the following items:

```
CVS/    mql/    src/    work/    Makefile
```

Here is a description of the contents of each:

- CVS/

  The CVS directory contains information relevant to the CVS archiving process. It should not be modified in any form by the user. All CVS maintenance should be accomplished via the standard CVS commands (see Appendix).

- src/

  The source directory has two subdirectories, ucb/ and uiuc/, which contain the Berkeley and UIUC analysis software, respectively. The UIUC code is currently "folded" into a MBAS module ("steven") for the purposes of online analysis. However, all of the MBAS software of concern resides in the ucb/ directory: header files, C++ modules, and MQL modules (see Appendix for a list of module descriptions).

- work/

  This is the destination for all compiled "source" programs. It is also the location of the final "mu" executable program (see the "Quickstart" section).

- mql/

  This is the location of the software supporting Fred's MQL language. It should not be modified by the user, but it *should* be periodically updated and recompiled (see Appendix).

- Makefile

  This is a top-level makefile which encompasses both the MQL support software and the actual analysis software. Note that it is possible (and recommended) to perform directory-specific "make" compilations, rather than comprehensive compilations. However, this top-level Makefile is useful when the mql/ directory has been updated and needs to be recompiled.

## 5.2 Customizing the MBAS for Personal Use

The simplest way to begin writing your own personal analysis code is to make a copy of the online.mql module in your own name. The following instructions will guide you in how to accomplish this:

- **Step 1 – Make a renamed copy of** online.mql

  In the directory mu/src/ucb, copy online.mql into a module of your own naming. For example,

  ```
  mu/work> cp online.mql username_spectra.mql
  ```

  Then, in the first line of your new personal MQL module, you need to change the title to reflect the new module name (the filename and module name must be identical).

- **Step 2 – Modify the MODULES list**

  In the file `mu/work/MODULES`, replace the entry `online` with the name of your new personal module, `username_spectra`.

- **Step 3 – Disable the `online` module in the Analyzer**

```
mu/work> odbedit
[ODBedit]> cd Analyzer/Module switches
[ODBedit ..switches]> set online n
[ODBedit ..switches]> quit
```

- **Step 4 – Rebuild the MBAS**

  In the directory `mu/work`, perform the following:

```
mu/work> make clean
mu/work> make
```

  And this will create the executable "mu".

From now on, you can edit your personal MQL file as you like. For instructions on using MQL, we refer you to Fred's MQL Manual at:

http://weak0.physics.berkeley.edu/weakint/research/muons/mql.pdf

The MQL language is quite easy to use, however, and you should be able to quickly learn a great deal by experimenting with changes to your personal MQL file. MQl was designed to allow users to compactly and efficiently perform complex coincidence calculations. We think that you will find it to be an attractive alternative to writing lengthy and convoluted C++ coincidence code.

We recommend that you keep copies of personalized modules in locations outside the MBAS `mu/` directory, in order to avoid inadvertant overwriting when CVS updates are performed.

# 6 Appendix

## 6.1 Words of Warning

Data analysis should *never* be performed on pc3608–**no exceptions**. There are two reasons for this restriction:

- pc3608 is our main DAQ server, and cannot afford to waste processor time on analysis
- risk of ODB corruption

The PSI clusters have more than enough computing power to serve all $\mu$Cap analysis needs. If you do not already have a PSI `afs` account, contact Claude Petitjean about obtaining one.

## 6.2 MBAS Module Descriptions

- `analyzer.cpp`

  The "master" MIDAS Analyzer program: it executes the MBAS modules in sequence, according to the order in which they are listed in the file `mu/work/MODULES`. The `analyzer.cpp` module should never be directly modified by the user; instead, the user should edit the `MODULES` file.

- `caen_comp.cpp`

  Moves through the CAEN and COMP raw data banks (CAE0,CAE1,CMP0,CMP1) and compiles a new bank, HITS, containing a list of detector hits by time and parameter number.

- `make_tdc400.cpp`

  Moves through the TDC400 raw data banks (TDC1-6) and creates absolute time banks, TD1-6, as well as track reconstructions for the 3 threshholds, TRK0-2.

- `sniffer.cpp`

  Moves through the track reconstruction bank TRK0 and "sniffs" for muon tracks. It compiles a bank of track candidates, SNIF, along with Bragg peak and strip hit information for each candidate.

- `raw_spectra.cpp`

  Fills histograms from raw data banks' information (e.g. hits on TPC anodes and cathodes for all threshholds, CAEN interpolator distributions, etc.);

- `online.mql`

  Fills histograms from processed data banks' information (HITS, SNIF). These are the spectra used for real-time analysis of $\mu$Cap data during run shifts.

## 6.3 Configuring the ODB

### 6.3.1 Setting Module Switches

Sometimes it is useful to disable certain MBAS analysis modules (for example, if you are interested in looking only at a run's CAEN data, there would be no reason to run the `make_tdc400` and `sniffer` modules, since they would only slow down the analysis). You can turn modules on and off using `odbedit`:

```
[ODBedit]> cd Analyzer/Module switches
[ODBedit ..switches]> ls
    caen_comp    y
    make_tdc400  y
    sniffer      y
    raw_spectra  y
    online       n
[ODBedit ..switches]> set online y
```

### 6.3.2 Turning ODB Loading On/Off

Each $\mu$Cap data file contains a copy of the DAQ ODB at the time the run was taken. It is possible to configure the Analyzer to either load the accompanying ODB with every run, or to ignore it and use the existing ODB:

```
[ODBedit]> cd Analyzer/
[ODBedit:Analyzer]> set ''ODB Load'' y
```

Unless the DAQ ODB was corrupted at the time of data-taking, it should be OK to set the "`ODB Load`" to "`y`".

## 6.4 CVS Maintenance

The analysis software is constantly being improved. Thus, you will need to periodically update the `mql/` and `src/` directories with the latest versions from the CVS archives. In order to do this, you must move into the desired directory and perform a CVS update:

```
mu/src> cvs update -d
```

This will replace older versions of the standard software with the most recent version.