

Near-Online data analysis

Brief User Guide to Beamtime DAQ Access

- [Brief User Guide to Beamtime DAQ Access](#)
 - [Introduction](#)
 - [TL;DR](#)
 - [Requirements](#)
 - [Access to the Maxwell cluster](#)
 - [Anaconda virtual environment](#)
 - [OS variants and Python Versions](#)
 - [Install the beamtimedaqaccess Anaconda package](#)
 - [BeamtimeDaqAccess for Matlab users](#)
 - [Start Matlab and Install the Library Wrapper](#)
 - [Confirm everything works with Matlab](#)
 - [Updating the Library](#)
 - [Access to the DESY software repository for library developers](#)
 - [Related resources](#)

Introduction

The library [BeamtimeDaqAccess](#) provides access to data acquired by experiments at FLASH. It retrieves data for a given DAQ channel and time period or DAQ run. At FLASH data files usually are stored on the [ASAP3 File System](#). They are accessible on the [Maxwell Cluster](#), workstations, and PCs at the beamline. This guide explains how to set-up and use the library on computers at FLASH. Of course the library [BeamtimeDaqAccess](#) could be used on any computer having access to the data. We provide Anaconda Python packages of the library for GNU/Linux and MS Windows. On GNU/Linux the library can also be used with Matlab.

TL;DR

```
alice@max-fsc:~$ module load anaconda3/5.2
alice@max-fsc:~$ source activate flash
(flash)alice@max-fsc:~$ conda install -c http://doocspkgs.desy.de/pub/flashconda beamtimedaqaccess
(flash)alice@max-fsc:~$ ipython
In [1]: import beamtimedaqaccess
In [2]: daq= beamtimedaqaccess.accessHdf("/pnfs/desy.de/flash1/disk/warehouse/data/hdf/online")
In [3]: adcl, ids= daq.allValuesOfRun("/FL1/Experiment/BL1/ADQ412 GHz ADC/CH00/TD", 26596, ":1000")
```

Requirements

Access to the Maxwell cluster

The [Maxwell Cluster](#) is a powerful farm of inter-connected PC nodes running Scientific GNU/Linux. To access the cluster you need a DESY AFS computing account or a so called scientific account.

- If you have a default DESY AFS account login to [max-fsc.desy.de](#) using your personal account. (From outside of DESY network tunnel through [bastion.desy.de](#) or use the [remote login](#) to [max-display.desy.de](#).)
- Scientific account users instead go to [desy-ps-cpu.desy.de](#). To connect to the Maxwell cluster from outside the DESY network -- eduroam or the DESY guest network are considered to be outside -- then you can SSH tunnel your connection through [desy-ps-ext.desy.de](#) or more conveniently directly use the [remote login](#) to [max-display.desy.de](#).

Note: Please, always do use your personal account to work on the Maxwell cluster or any central DESY computing resources. Don't use the functional accounts like vuvfuser or bl1user. Using functional accounts on shared resources causes many difficult to predict problems, even if it might appear to work for you in the beginning. Also, **never** try to login to [bastion](#), [desy-ps-ext](#), or [max-display](#) using one of the functional accounts.

The rest of this guide assumes you can login to the suggested Maxwell cluster node and are able to successfully run the command `xclock` to get a clock face displayed on your client computer.

Anaconda virtual environment

For Python we strongly suggest to use [Anaconda virtual environments](#) on the DESY computers. Even if you intend to use [BeamtimeDaqAccess](#) with Matlab do use Anaconda Python and create your virtual environment as detailed on the page [Anaconda Python at FLASH](#).

On Maxwell you need to load the Anaconda3 module first:

```
$ module load anaconda3/5.2
```

On the workstations and user consoles in the FLASH halls, Anaconda is [installed and available](#) at `/opt/conda/bin/`. Add this to your shell's search path, if it is not on the path already. In bash:

```
if [[ ":$PATH:" != *"/opt/conda/bin:*" ]]; then PATH=$PATH:/opt/conda/bin ; fi
```

So, do create your Anaconda virtual environment and activate it as explained in detail on the page [Anaconda virtual environments](#):

```
$ conda create -n flash python=3.7 ipykernel
$ source activate flash
	flash)$
```

OS variants and Python Versions

The library is compatible with Python 2.7.5 and newer, Python 3.4 and newer on GNU/Linux, Mac OS 10 and MS Windows. We provide Anaconda Python packages for easy installation for Python 2.7, 3.5 (deprecated) 3.6, and 3.7 for GNU/Linux and MS Windows. On GNU/Linux the library can also be used with Matlab versions R2016b and newer.

Install the beamtimedaqaccess Anaconda package

To install the library BeamtimeDaqAccess into your active Anaconda virtual environment execute

```
(flash)$ conda install -c http://doocspkgs.desy.de/pub/flashconda beamtimedaqaccess
```

and you are ready to go. The following example uses Ipython:

```
(flash)$ ipython
In [1]: import beamtimedaqaccess
In [2]: daq= beamtimedaqaccess.accessHdf("/pnfs/desy.de/flash1/disk/warehouse/data/hdf/online")
In [3]: adcl, ids= daq.allValuesOfRun("/FL1/Experiment/BL1/ADQ412 GHz ADC/CH00/TD", 26596, ":1000")
```

The details how to use and install Anaconda environments for software at FLASH can be found on the page [Anaconda virtual environments](#).

BeamtimeDaqAccess for Matlab users

Note, we no longer explicitly support Matlab. However, we still maintain a Matlab wrapper to use Matlab's Python interface for BeamtimeDaqAccess on GNU/Linux.

Start Matlab and Install the Library Wrapper

To use Matlab on Maxwell you need to add the corresponding Module, e.g.:

```
$ module add matlab/R2018b
```

On the workstations and user consoles in the FLASH halls, flashxuser1 and 2 Matlab R2016b is installed by default.

The most painless way to use BeamtimeDaqAccess with Matlab is to set-up an Anaconda virtual environment. The beamtimedaqaccess package can be installed to the environment as shown in the previous section. Then start Matlab from within the active virtual environment like so:

```
(flash)$ LD_PRELOAD=$CONDA_PREFIX/lib/libhdf5.so matlab -nodesktop
>>
```

The `LD_PRELOAD` is required to use the hdf5 library with Python. It however precludes the use of `h5read` et al. at the same time in one Matlab session. The option `-nodesktop` is optional.

To use the BeamtimeDaqAccess library wrapper, you need to install it once. In the Matlab command window change to your Matlab userpath, download the library, and change back to the original working directory:

```
>> originalDir= cd(userpath)
>> !curl -O https://stash.desy.de/projects/CS/repos/pah/raw/matlab/BeamtimeDaqAccess.m
>> cd(originalDir)
>> help BeamtimeDaqAccess
```

Only if you have set the Matlab Python version before, you need to change this persistent setting once again. In the Matlab command window:

```
>> pyversion([getenv('CONDA_PREFIX') '/bin/python']);
```

Confirm everything works with Matlab

To check everything works as expected use our example HDF files available on Maxwell and the workstations in the hall or flashluser1 and 2:

```
>> daq= BeamtimeDaqAccess('/pnfs/desy.de/flashl/disk/warehouse/data/hdf/online');
>> adcDset= ('/FL1/Experiment/BL1/ADQ412 GHz ADC/CH00/TD');
>> run= 26596;
>> [adcTrace, trainIds]= daq.allValuesOfRun(adcDset, run, '100000:120000');
>> imagesc(adcTrace);
```

If Matlab crashes, you might have forgotten to start Matlab using the Bash shell with the LD_PRELOAD set as shown above (sorry, we give examples for the Bash shell only).

Updating the Library

The BeamtimeDaqAccess library is under development and many improvements and fixes are released even during the beamtime. Therefore, from time to time you will be asked to update the library.

In the Bash shell with your chosen Anaconda environment activated:

```
(flash)$ conda update -c http://doocspkgs.desy.de/pub/flashconda beamtimedaqaccess
```

Matlab users in addition might need to update the Wrapper. In the Matlab command window:

```
>> cd(userpath)
>> !curl -O https://stash.desy.de/projects/CS/repos/pah/raw/matlab/BeamtimeDaqAccess.m
```

Access to the DESY software repository for library developers

The BeamtimeDaqAccess library is kept on the central DESY software revision control system [DESY Bitbucket](#). The [PAH project's repository](#) is publicly available. To browse its contents click on this [link](#).

If you want to contribute to the PAH project, you need registered access to [DESY Bitbucket](#). To find out if you have access go to the [the Bitbucket landing page](#) and try to login using your DESY computing or scientific account user ID and password. If you can not login, apply at our group administrators or fs-ec@desy.de for the resource "_stash_".

Related resources

- [Matlab Beamtime DAQ Access Library Doc](#)
- [Python Beamtime DAQ Access Library Doc](#)
- [Maxwell for Photon Science](#)