Hands on BASF2/GBASF2

Belle Analysis Workshop, 2024

BASF2

- **BASF2**- Belle II Analysis Software Framework

The primary software framework used for simulating, reconstructing, and analyzing data for the Belle II experiment

Key features of **BASF2**:

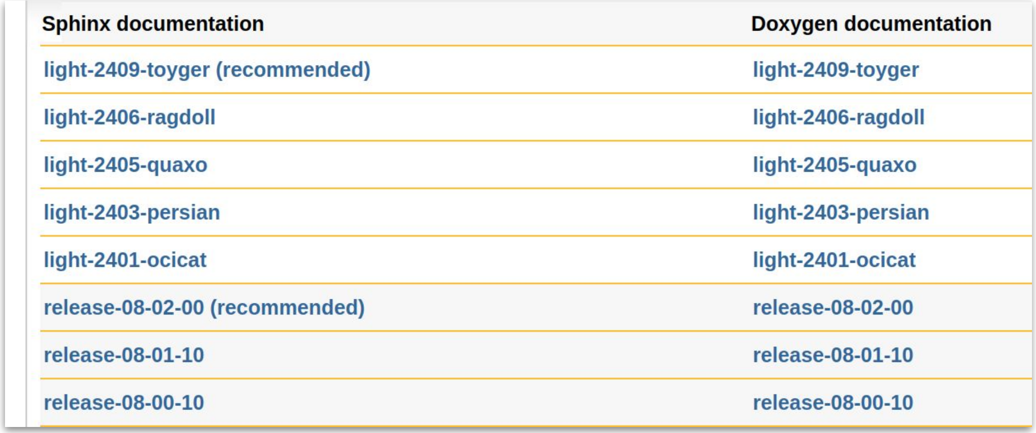
- **Modular Design**: It consists of different modules, each performing specific tasks like event generation, reconstruction, or analysis.
- **Python Interface**: Users write scripts in Python to create and configure processing chains, which are then executed by the C++ backend.
- **Simulation and Reconstruction**: It handles everything from simulating particle collisions to reconstructing the resulting particle trajectories and decays.
- **ROOT Integration**: It is tightly integrated with ROOT, a data analysis framework widely used in high-energy physics, allowing for easy handling of histograms, trees, and other data structures.

BASF2

- BASF2- Belle II Analysis Software Framework

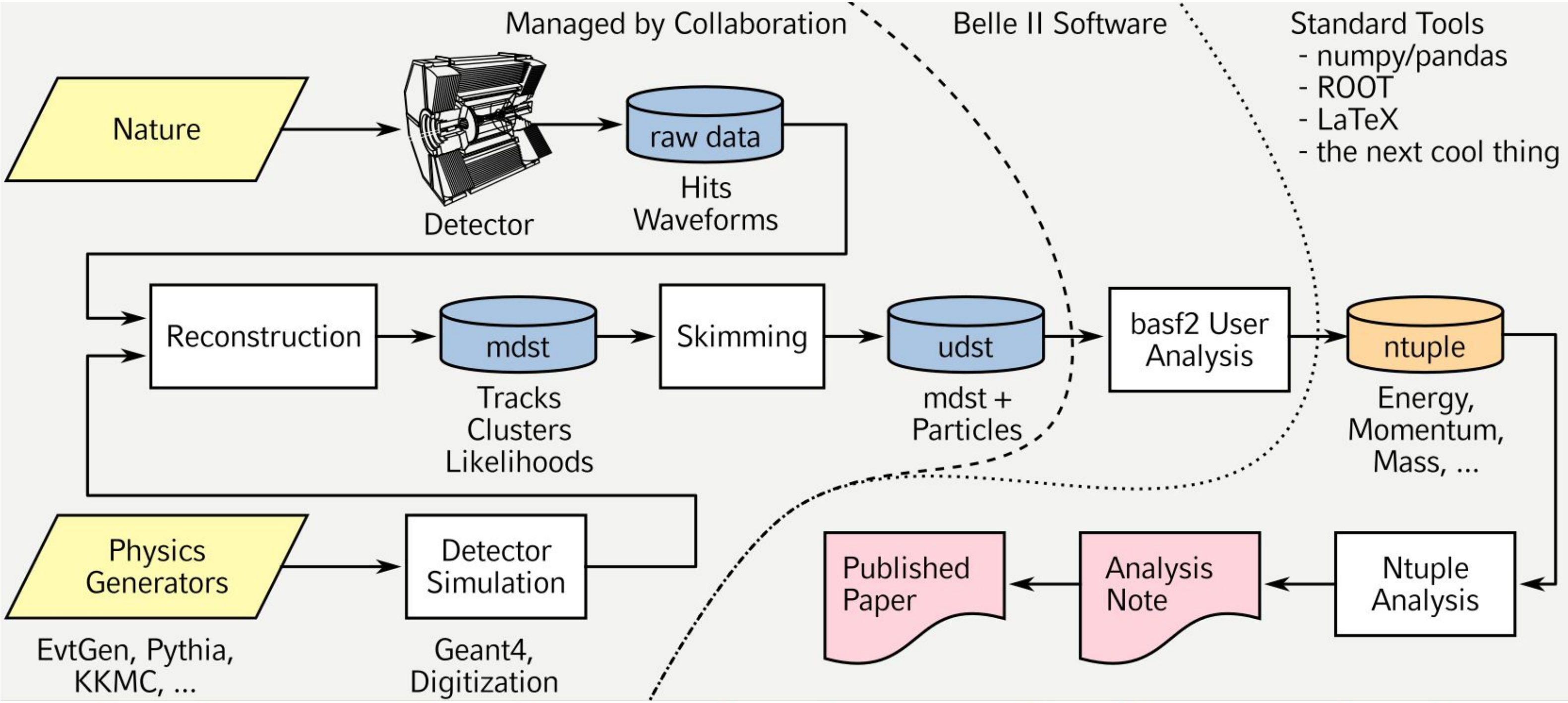
- source code: https://github.com/belle2/basf2
- **Documentation: https://software.belle2.org/**
- Basf2 links against defined set of third-party libraries (externals): https://github.com/belle2/externals
- Install and set up basf2 called tools: https://github.com/belle2/tools
- Script for version managing (recommended releases and global tags): https://github.com/belle2/versioning

Ask your doubt at: https://questions.belle2.org/



Sphinx documentation	Doxygen documentation
light-2409-toyger (recommended)	light-2409-toyger
light-2406-ragdoll	light-2406-ragdoll
light-2405-quaxo	light-2405-quaxo
light-2403-persian	light-2403-persian
light-2401-ocicat	light-2401-ocicat
release-08-02-00 (recommended)	release-08-02-00
release-08-01-10	release-08-01-10
release-08-00-10	release-08-00-10

BASF2



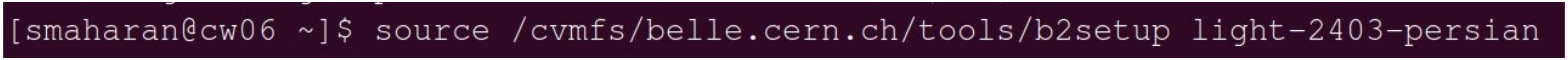
Load input data file → Filter out certain particles → Reconstruct decay(s) → Apply fits → Match with MC → Save variables to file

Source BASF2

- source /cvmfs/belle.cern.ch/tools/b2setup <basf2_release>

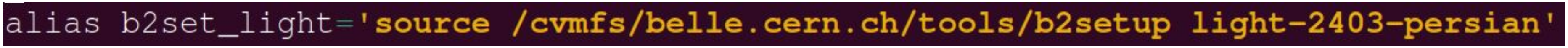
basf2 Releases

1. Main releases (release-major-minor-patch, eg: release-08-01-06)
- Includes all approved changes and suitable for everything, data-taking, analysis, MC data production, etc.
2. Light releases (light-yymm-cat breed, eg: light-2405-quaxo)
- Includes only libraries necessary for analysis, updated with most bug fixes and features
- Designed for high-level analysis only (can't process anything other than mdst/udst files)

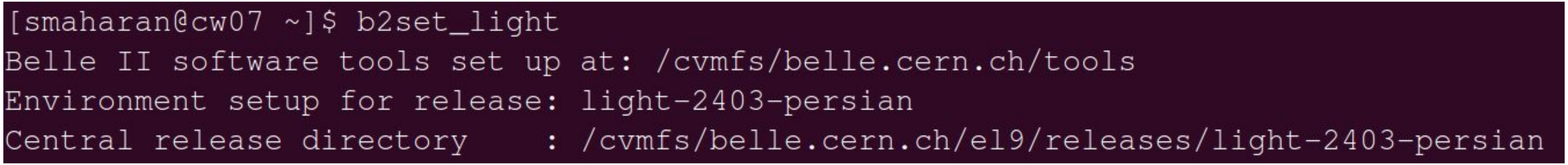


```
[smaharan@cw06 ~]$ source /cvmfs/belle.cern.ch/tools/b2setup light-2403-persian
```

Or in your ~/.bashrc add-



```
alias b2set_light='source /cvmfs/belle.cern.ch/tools/b2setup light-2403-persian'
```



```
[smaharan@cw07 ~]$ b2set_light
Belle II software tools set up at: /cvmfs/belle.cern.ch/tools
Environment setup for release: light-2403-persian
Central release directory    : /cvmfs/belle.cern.ch/el9/releases/light-2403-persian
```

Prerequisite

Before you get started with our script, you need

1. Active KEK computing account
   - ssh kekcc
2. Active DESY account
   - Cloning from gitlab
3. Basic knowledge of Linux, git, python, Jupyter notebook, matplotlib, NumPy, pandas

Let’s start the analysis

- We will reconstruct-

  $B^{+} \rightarrow K^{*+}[\,]J/\psi[\,]$

Let’s start the analysis

- We will reconstruct-

$B^+ \to K^{*+}[\to K^+\pi^0]J/\psi[\to \mu^+\mu^-]$
$B^+ \to K^{*+}[\to K^+\pi^0]J/\psi[\to e^+e^-]$
$B^+ \to K^{*+}[\to K_S{}^0\pi^+]J/\psi[\to \mu^+\mu^-]$
$B^+ \to K^{*+}[\to K_S{}^0\pi^+]J/\psi[\to e^+e^-]$

Please find the file-
/home/belle2/smaharan/BAW_tutorial/BtoJpsiks_tutorial.py

Basic structure

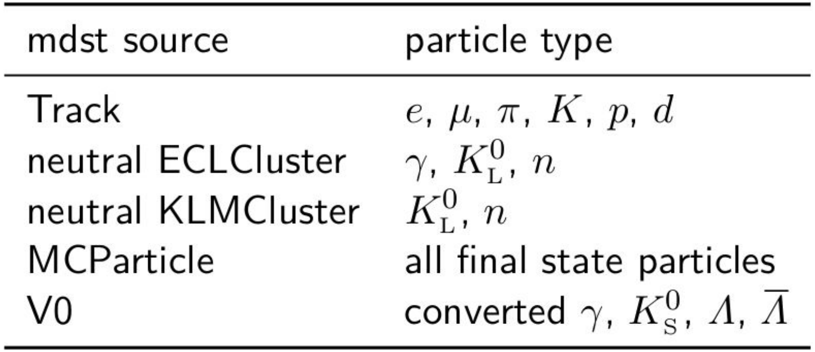
- import basf2, modularAnalysis, other packages
- Start with a path, main = basf2.Path()
- Specify input MDST file: inputMdst, inputMdstList
- Load/fill particle lists: fillParticleList, fillParticleLists
- Apply any cuts, if needed: applyCuts
- If electrons are in the final state particle list, consider Bremsstrahlung correction: correctBrems
- Build Event Kinematics, if needed: buildEventKinematics
- Start reconstructing your decay chain: reconstructDecay
- Apply cuts, if needed: applyCuts
- Match MC Truth: matchMCTruth
- Apply vertex fit: K-fit or tree fit, if needed: vertex.treefit
- If multiple reconstructed particles, apply best candidate selection (BCS): rankByHighest
- Build Rest of Event, append ROE mask, apply any necessary cuts: buildRestOfEvent, appendROEMask
- Write out the variables as Ntuples to the output root file: VariablesToNtuple
- Execute path: basf2.process(main)

Particles

- fillParticleList('pi+:all', cut=“”, path)
  creates two ParticleLists: 'pi+:all' with all positively charged pions and 'pi-:all' with all negatively charged pions.
- So what when you do fillParticleList('K-:all', cut=“”, path)
  each track fitted with up to six mass hypotheses (at least one track must have converged)
- stdKshorts(prioritiseV0=True, fitter='TreeFit', path)
  creates K_S0:merged and
- stdLambdas(prioritiseV0=True, fitter='TreeFit', path)
  creates Lambda0:merged

(vertex t methods “KFit”, “TreeFit”, and “Rave” available)

- stdXi(fitter='TreeFit', path) creates Xi-:std
- stdOmega(fitter='TreeFit', path) creates Omega-:std
- stdPi0s(listtype='eff60_May2020', path) creates π 0 list with 60 % signal eciency (check confluence)
- No recommended predened standard particle lists or charged hadron nal state particles



mdst source	particle type
Track	e, μ, π, K, p, d
neutral ECLCluster	γ, K^0_{L}, n
neutral KLMCluster	K^0_{L}, n
MCParticle	all final state particles
V0	converted γ, K^0_{S}, Λ, $\overline{\Lambda}$

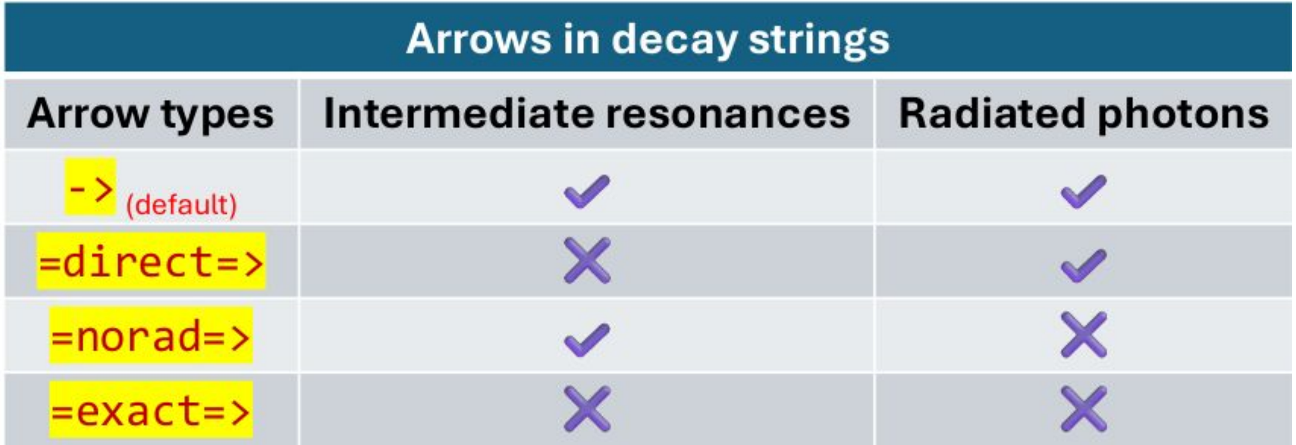
Marker in decay string

- ^ : selection of succeeding particle
- @ : succeeding particle is unspecified, useful or inclusive reconstructions
- ... : urther massive particles in decay mode possible
- ?nu : decay mode might contain a neutrino
- ?gamma : decay mode might contain radiative photons
- ?addbrems : decay mode might contain bremsstrahlung photons
- (misID) : succeeding particle is allowed to be other particle type
- (decay) : succeeding particle might have decayed in fight, e.g. $\pi \to \mu\nu\mu$

Decay string syntax

Commonly used with reconstructDecay() and for creating aliases

- “Mother particle” arrow “daughter particle(s)”: D0:kpi -> K-:loose pi+:loose
- To construct a decay sequence, use square brackets: D*+ -> [D0 -> K- pi+] pi+:slow



Arrows in decay strings

Arrow types	Intermediate resonances	Radiated photons
-> (default)	✓	✓
=direct=>	✗	✓
=norad=>	✓	✗
=exact=>	✗	✗

Different arrows are allowed in same decay str, D*+ -> [D0 =direct=> K- pi+ pi0] pi+

Run your script

- Run your script using-
  basf2 [OPTIONS] [STEERING_FILE] [-- [STEERING_FILE_OPTIONS]]

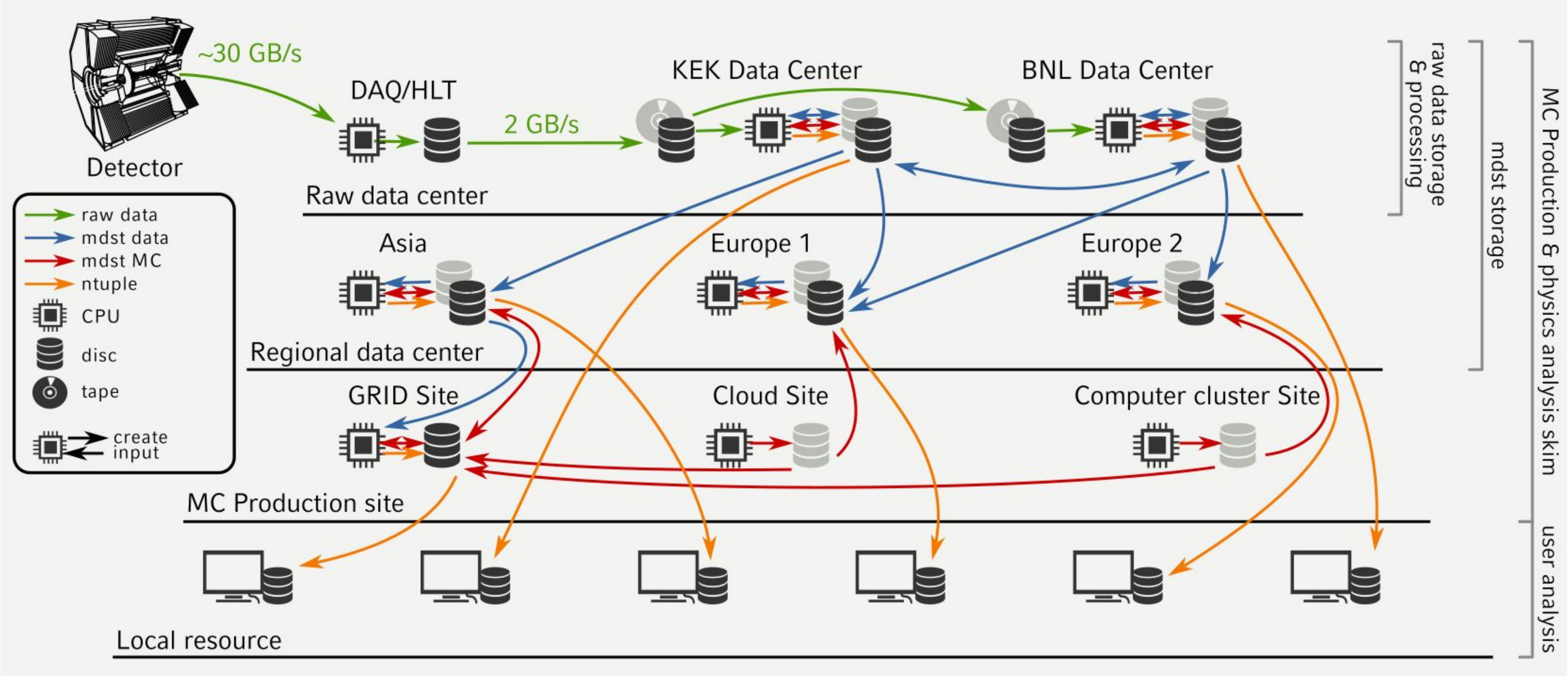
- Some useful command-line options [OPTIONS]:
  NOTE: THESE OPTIONS HAVE HIGHER PRIORITY THAN THOSE IN YOUR STEERING FILE
- --dry-run : Useful for checking errors, etc., it doesn't actually execute the Path over various events
- -n <N> : Limits execution to N events.
- -i : Specify the input filename.
- -o : Specify the output filename.
- --help : Prints full list of available command-line options

GBASF2

Welcome! You are ready

for physics analysis........

GBASF2


~30 GB/s
Detector
DAQ/HLT
2 GB/s
KEK Data Center
BNL Data Center
Raw data center
Asia
Europe 1
Europe 2
Regional data center
GRID Site
Cloud Site
Computer cluster Site
MC Production site
Local resource
raw data
mdst data
mdst MC
ntuple
CPU
disc
tape
create
input
raw data storage & processing
mdst storage
MC Production & physics analysis skim
user analysis

GBASF2

Prerequisites

- Valid Grid certificate issued within one year on the ~/.globus and in the browser.
- Make sure to have rw permission of the userkey.pem for the user, not anyone else.
- Well tested working basf2 reconstruction script.

Important Links

- GarudaIndia
- gbasf2 documentation
- Belle dirac

Setup preinstalled gbasf2

#use a different terminal where basf2 is not set up already

```
$  source /cvmfs/belle.kek.jp/grid/gbasf2/pro/bashrc
gb2_proxy_init -g belle
Generating proxy...
Enter Certificate password: *************
```

GBASF2

Look at the available gb2 tools

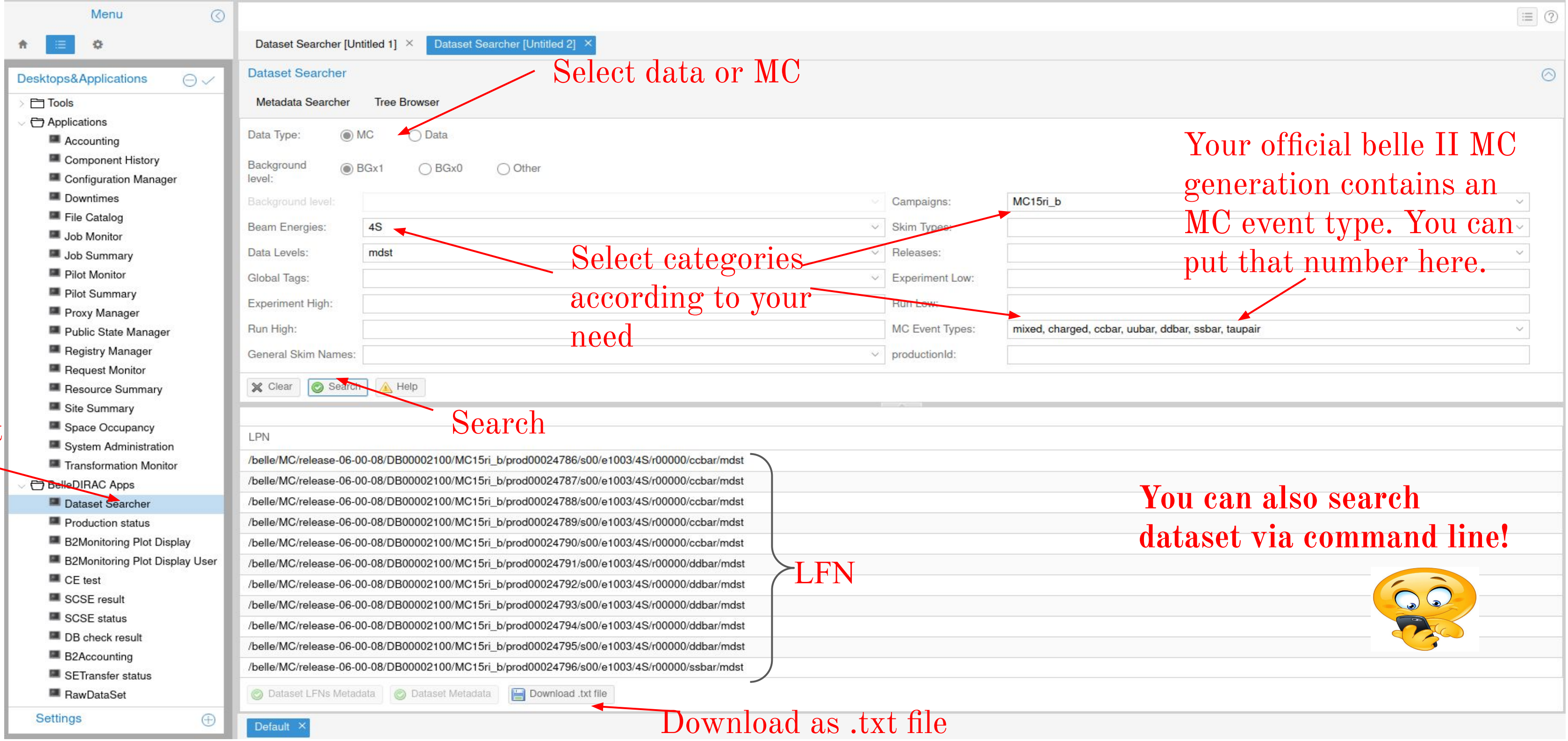
```
$ gb2_<tab><tab>
gb2_check_release      gb2_ds_get               gb2_ds_quota              gb2_ds_siteForecast
gb2_job_reschedule     gb2_prod_dataset         gb2_proxy_destroy         gb2_diagnostic
gb2_ds_list            gb2_ds_rep               gb2_ds_sync               gb2_job_status
gb2_prod_releases      gb2_proxy_info           gb2_ds_collection         gb2_ds_put
gb2_ds_rep_status      gb2_job_delete           gb2_list_destse           gb2_prod_status
gb2_proxy_init         gb2_ds_count_events      gb2_ds_query_datablock    gb2_ds_rm
gb2_job_kill           gb2_list_site            gb2_prod_summary          gb2_se_list
gb2_ds_du              gb2_ds_query_dataset     gb2_ds_rm_rep             gb2_job_output
gb2_prod_accounted     gb2_project_analysis     gb2_update                gb2_ds_generate
gb2_ds_query_file      gb2_ds_search            gb2_job_parameters        gb2_prod_campaigns
gb2_project_summary
```

Checkout more info

```
$  gb2_check_release [-h] [-v] [--usage]
```

GBASF2

Search dataset on Belle dirac



Data Type: ● MC ○ Data

Background level: ● BGx1 ○ BGx0 ○ Other

Background level:

Beam Energies: 4S

Data Levels: mdst

Global Tags:

Experiment High:

Run High:

General Skim Names:

Campaigns: MC15ri_b

Skim Types:

Releases:

Experiment Low:

Run Low:

MC Event Types: mixed, charged, ccbar, uubar, ddbar, ssbar, taupair

productionId:

LPN

/belle/MC/release-06-00-08/DB00002100/MC15ri_b/prod00024786/s00/e1003/4S/r00000/ccbar/mdst
/belle/MC/release-06-00-08/DB00002100/MC15ri_b/prod00024787/s00/e1003/4S/r00000/ccbar/mdst
/belle/MC/release-06-00-08/DB00002100/MC15ri_b/prod00024788/s00/e1003/4S/r00000/ccbar/mdst
/belle/MC/release-06-00-08/DB00002100/MC15ri_b/prod00024789/s00/e1003/4S/r00000/ccbar/mdst
/belle/MC/release-06-00-08/DB00002100/MC15ri_b/prod00024790/s00/e1003/4S/r00000/ccbar/mdst
/belle/MC/release-06-00-08/DB00002100/MC15ri_b/prod00024791/s00/e1003/4S/r00000/ddbar/mdst
/belle/MC/release-06-00-08/DB00002100/MC15ri_b/prod00024792/s00/e1003/4S/r00000/ddbar/mdst
/belle/MC/release-06-00-08/DB00002100/MC15ri_b/prod00024793/s00/e1003/4S/r00000/ddbar/mdst
/belle/MC/release-06-00-08/DB00002100/MC15ri_b/prod00024794/s00/e1003/4S/r00000/ddbar/mdst
/belle/MC/release-06-00-08/DB00002100/MC15ri_b/prod00024795/s00/e1003/4S/r00000/ddbar/mdst
/belle/MC/release-06-00-08/DB00002100/MC15ri_b/prod00024796/s00/e1003/4S/r00000/ssbar/mdst

Go to Dataset searcher

Select data or MC

Your official belle II MC generation contains an MC event type. You can put that number here.

Select categories according to your need

Search

LFN

You can also search dataset via command line!

Download as .txt file

GBASF2

Submit job with input files

```
$ gbasf2 <steering_file> -p <project_name> -s <basf2_release> -i <input_LFN>
```

Submit job with input filelist

```
$ gbasf2 <steering_file> -p myproject -s release --input_dslist <LFN_list>
```

Submit job with your own module

```
$ gbasf2 <steering_file> -f filename
```

Example

```
$ gbasf2 BtoJpsiks_tutorial.py -p gbasf2BAW2024 -s light-2409-toyger -i
/belle/MC/release-06-00-08/DB00002100/MC15ri_b/prod00024816/s00/e1003/4S/r00000/charged/mdst
/sub00/mdst_000001_prod00024816_task10020000001.root
**********************************************
*************** Project summary ****************
** Project name: gbasf2BAW2024
** Dataset path: /belle/user/csourabh/gbasf2BAW2024
** Steering file: analysis_jpsiks.py
** Job owner: csourabh @ belle (23:48:19)
** Preferred site / SE: None / None
** Input files for first job: LFN:/belle/MC/release-06-00-08/DB00002100/MC15ri_b/prod00024816/s00/e1003/4S/r00000/charged/mdst/sub00/mdst_000001_prod00024816_task10020000001.root
** Number of input files: 600
** Number of jobs: 600
** Processed data (MB): 1149971
** Processed events: 108000000 events
** Estimated CPU time per job: 3000 min
**********************************************
Are you sure to submit the project?
Please enter Y or N:
```

Put Y to submit the jobs. You did it. Congratulations...

GBASF2

Monitor Jobs

```
$ gb2_project_summary
```

For more information

```
$ gb2_job_status -p <project name>
```

Check output info of a failed job

```
$ gb2_job_output -j <job id>
```

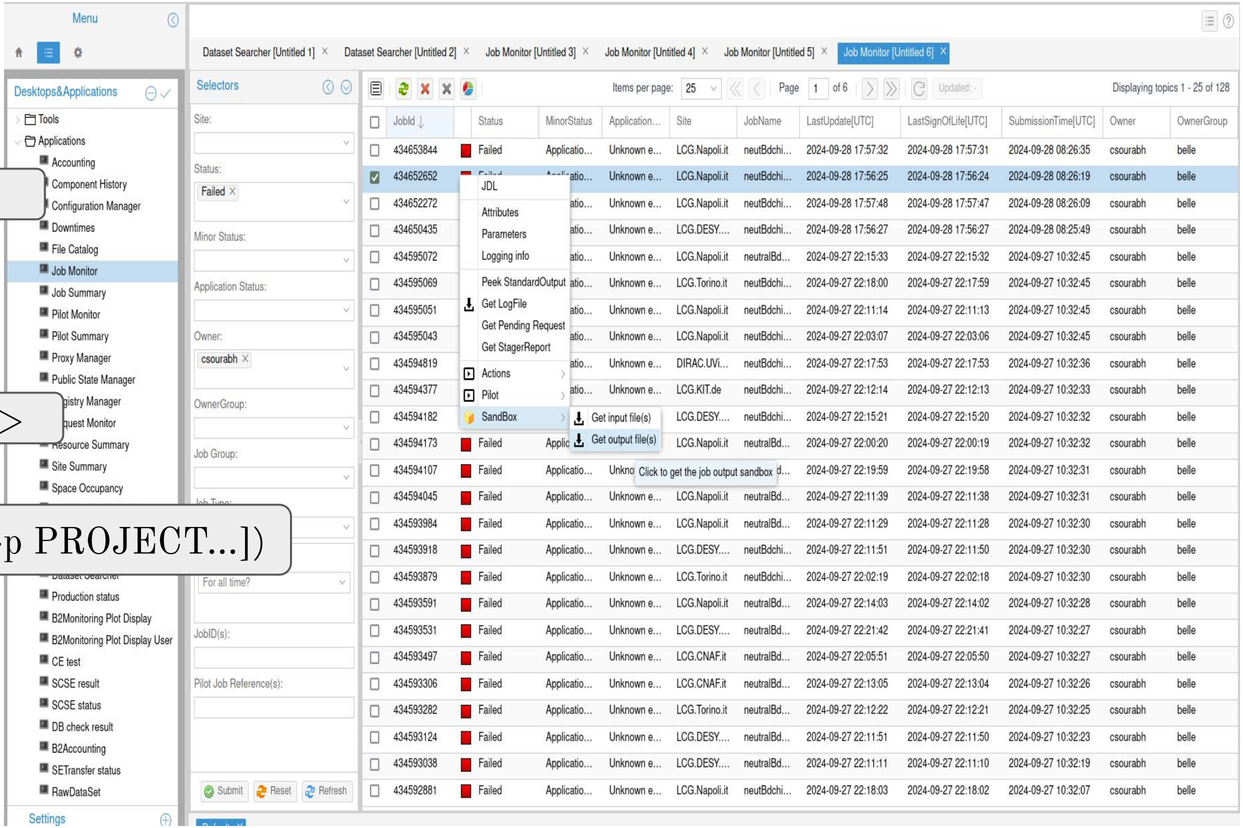
For more information about a failed job

```
$ gb2_diagnostic --failed_job <job id>
```

Reschedule a failed job/project

```
$ gb2_job_reschedule ([-j JOBID... | -p PROJECT...])
```

Similarly, use diagnostic tools for the jobs waiting for a long time and the jobs failed during download.



Displaying topics 1 - 25 of 128

☐	JobId		Status	MinorStatus	Application...	Site	JobName	LastUpdate[UTC]	LastSignOfLife[UTC]	SubmissionTime[UTC]	Owner	OwnerGroup
☐	434653844	🟥	Failed	Applicatio...	Unknown e...	LCG.Napoli.it	neutBdchi...	2024-09-28 17:57:32	2024-09-28 17:57:31	2024-09-28 08:26:35	csourabh	belle
☑	434652652			atio...	Unknown e...	LCG.Napoli.it	neutBdchi...	2024-09-28 17:56:25	2024-09-28 17:56:24	2024-09-28 08:26:19	csourabh	belle
☐	434652272			atio...	Unknown e...	LCG.Napoli.it	neutBdchi...	2024-09-28 17:57:48	2024-09-28 17:57:47	2024-09-28 08:26:09	csourabh	belle
☐	434650435			atio...	Unknown e...	LCG.DESY....	neutBdchi...	2024-09-28 17:56:27	2024-09-28 17:56:27	2024-09-28 08:25:49	csourabh	belle
☐	434595072			atio...	Unknown e...	LCG.Napoli.it	neutralBd...	2024-09-27 22:15:33	2024-09-27 22:15:32	2024-09-27 10:32:45	csourabh	belle
☐	434595069			atio...	Unknown e...	LCG.Torino.it	neutBdchi...	2024-09-27 22:18:00	2024-09-27 22:17:59	2024-09-27 10:32:45	csourabh	belle
☐	434595051			atio...	Unknown e...	LCG.Napoli.it	neutBdchi...	2024-09-27 22:11:14	2024-09-27 22:11:13	2024-09-27 10:32:45	csourabh	belle
☐	434595043			atio...	Unknown e...	LCG.Napoli.it	neutBdchi...	2024-09-27 22:03:07	2024-09-27 22:03:06	2024-09-27 10:32:45	csourabh	belle
☐	434594819			atio...	Unknown e...	DIRAC.UVi...	neutBdchi...	2024-09-27 22:17:53	2024-09-27 22:17:53	2024-09-27 10:32:36	csourabh	belle
☐	434594377			atio...	Unknown e...	LCG.KIT.de	neutBdchi...	2024-09-27 22:12:14	2024-09-27 22:12:13	2024-09-27 10:32:33	csourabh	belle
☐	434594182					LCG.DESY....	neutBdchi...	2024-09-27 22:15:21	2024-09-27 22:15:20	2024-09-27 10:32:32	csourabh	belle
☐	434594173	🟥	Failed	Applic		LCG.Napoli.it	neutralBd...	2024-09-27 22:00:20	2024-09-27 22:00:19	2024-09-27 10:32:32	csourabh	belle
☐	434594107	🟥	Failed	Applicatio...	Unkno		d...	2024-09-27 22:19:59	2024-09-27 22:19:58	2024-09-27 10:32:31	csourabh	belle
☐	434594045	🟥	Failed	Applicatio...	Unknown e...	LCG.Napoli.it	neutralBd...	2024-09-27 22:11:39	2024-09-27 22:11:38	2024-09-27 10:32:31	csourabh	belle
☐	434593984	🟥	Failed	Applicatio...	Unknown e...	LCG.Napoli.it	neutralBd...	2024-09-27 22:11:29	2024-09-27 22:11:28	2024-09-27 10:32:30	csourabh	belle
☐	434593918	🟥	Failed	Applicatio...	Unknown e...	LCG.DESY....	neutBdchi...	2024-09-27 22:11:51	2024-09-27 22:11:50	2024-09-27 10:32:30	csourabh	belle
☐	434593879	🟥	Failed	Applicatio...	Unknown e...	LCG.Torino.it	neutBdchi...	2024-09-27 22:02:19	2024-09-27 22:02:18	2024-09-27 10:32:30	csourabh	belle
☐	434593591	🟥	Failed	Applicatio...	Unknown e...	LCG.Napoli.it	neutralBd...	2024-09-27 22:14:03	2024-09-27 22:14:02	2024-09-27 10:32:28	csourabh	belle
☐	434593531	🟥	Failed	Applicatio...	Unknown e...	LCG.DESY....	neutralBd...	2024-09-27 22:21:42	2024-09-27 22:21:41	2024-09-27 10:32:27	csourabh	belle
☐	434593497	🟥	Failed	Applicatio...	Unknown e...	LCG.CNAF.it	neutralBd...	2024-09-27 22:05:51	2024-09-27 22:05:50	2024-09-27 10:32:27	csourabh	belle
☐	434593306	🟥	Failed	Applicatio...	Unknown e...	LCG.CNAF.it	neutralBd...	2024-09-27 22:13:05	2024-09-27 22:13:04	2024-09-27 10:32:26	csourabh	belle
☐	434593282	🟥	Failed	Applicatio...	Unknown e...	LCG.Torino.it	neutralBd...	2024-09-27 22:12:22	2024-09-27 22:12:21	2024-09-27 10:32:25	csourabh	belle
☐	434593124	🟥	Failed	Applicatio...	Unknown e...	LCG.DESY....	neutralBd...	2024-09-27 22:11:51	2024-09-27 22:11:50	2024-09-27 10:32:23	csourabh	belle
☐	434593038	🟥	Failed	Applicatio...	Unknown e...	LCG.DESY....	neutralBd...	2024-09-27 22:11:11	2024-09-27 22:11:10	2024-09-27 10:32:19	csourabh	belle
☐	434592881	🟥	Failed	Applicatio...	Unknown e...	LCG.Napoli.it	neutBdchi...	2024-09-27 22:18:03	2024-09-27 22:18:02	2024-09-27 10:32:07	csourabh	belle

This can be done in dirac page as well.

GBASF2

After the jobs are completed, with good status

```
$ gb2_ds_list -u username
$ gb2_ds_list dataset -s all
```

By default gb2_ds_list use your username and lists the good jobs. But you can list other jobs also.

Download project

```
$ gb2_ds_get <project_name>
```

Clean out grid spaces after the jobs are downloaded

```
$ gb2_ds_rm <project name>
```

If you get lost, please mail the experts:
comp-users-forum@belle2.org

Try out by yourself: gbasf2 documentation

- Different positional arguments
- Search dataset via command line
- Setting up CPU time
- Replica Management
- Local gbasf2 installation(not necessary)

Arigatou gozaimasu
(ありがとう ございます)