



GBASF2とグリッド上のデータ

早坂 圭司 (新潟大学)

最初に…

- グリッド証明書を取得しましたか？（まだ→手遅れ）
- VOMS サーバに行ってBelle VO登録しましたか？（まだ→手遅れ）
- DIRACにユーザ登録しましたか？（今なら…加藤さんが？）

今回は…

- グリッドのオーバービュー的なことは行いません。
- 用語も結構適当です。特に「グリッド」
→正しくは「Belle II 分散コンピューティングシステム」です。
- ミニマムな知識でごまかし、ごまかし…
- 2015年のグリッド講習会の三宅さんのスライドがその辺はフォローしてくれてますので、後で見てください。

<https://agenda.hepl.phys.nagoya-u.ac.jp/indico/getFile.py/access?contribId=3&resId=0&materialId=slides&confId=210>

basf2とgbasf2

- gbasf2は、グリッド上でbasf2を実行してくれる、「と思って
くれて」十分です。

basf2 example.py

ローカルでexample.py

を実行

gbasf2 example.py

グリッド上でexample.py

を実行

とは言え、なんにもかもbasf2と同じ、というわけにはいきません。

まずは環境設定から…

準備その 1

- usercert.pem, userkey.pemを~/globusにコピーしましょう。
 - まず mkdir ~/globus
 - 次にこれらのファイルがあるディレクトリに移動して
 - cp usercert.pem userkey.pem ~/globus
 - 次にこれらのファイルの属性をチェック

```
[kiyohaya@cw02 tutorial20170624]$ ls ~/globus/ -la
total 0
drwxr-xr-x  2 kiyohaya b_belle 4096 May  8 19:01 ./
drwxr-x---+ 17 kiyohaya b_belle 4096 Jun 17 04:13 ../
-rw-r--r--  1 kiyohaya b_belle 1610 Jan 23 21:23 usercert.pem
-r-----  1 kiyohaya b_belle 1751 Jan 23 21:23 userkey.pem
```

userkey.pemがr-----になっていることが重要

なってない人は、cd ~/globusして、chmod 400 userkey.pem

準備その2 (gbasf2の準備 1)

```
[kiyohaya@cw11 tutorial20170624]$ mkdir gbasf2KEK && cd gbasf2KEK
[kiyohaya@cw11 gbasf2KEK]$ wget -N https://raw.githubusercontent.com/DIRACGrid/DIRAC/v6r17/
Core/scripts/dirac-install.py --no-check-certificate
--2017-06-17 09:06:48-- https://raw.githubusercontent.com/DIRACGrid/DIRAC/v6r17/Core/scrip
ts/dirac-install.py
Resolving raw.githubusercontent.com... 151.101.72.133
Connecting to raw.githubusercontent.com|151.101.72.133|:443... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://raw.githubusercontent.com/DIRACGrid/DIRAC/v6r17/Core/scripts/d
irac-install.py [following]
--2017-06-17 09:06:49-- https://raw.githubusercontent.com/DIRACGrid/DIRAC/v6r17
/Core/scripts/dirac-install.py
Resolving raw.githubusercontent.com... 151.101.72.133
Connecting to raw.githubusercontent.com|151.101.72.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 60553 (59K) [text/plain]
Saving to: "dirac-install.py"

100%[=====>] 60,553      ---K/s   in 0.006s

Last-modified header missing -- time-stamps turned off.
2017-06-17 09:06:50 (9.61 MB/s) - "dirac-install.py" saved [60553/60553]
```

```
wget -N https://raw.githubusercontent.com/DIRACGrid/DIRAC/v6r17/Core/scripts/dirac-install.py --no-check-certificate
```

準備その3 (gbasf2の準備 2)

```
python dirac-install.py -V Belle-KEK
```

```
[kiyohaya@cw11 gbasf2KEK]$ python dirac-install.py -V Belle-KEK
2017-06-17 00:10:35 UTC dirac-install [NOTICE] Processing installation requirements
2017-06-17 00:10:36 UTC dirac-install [NOTICE] Destination path for installation is /gpfs/home/belle/kiyohaya/tutorial20170624/gbasf2KEK
```

いろいろ出てきます…

```
2017-06-17 00:10:52 UTC dirac-install [NOTICE] Executing /gpfs/home/belle/kiyohaya/tutorial20170624/gbasf2KEK/scripts/dirac-externals-requirements...
2017-06-17 00:10:54 UTC dirac-install [NOTICE] Belle-KEK properly installed
```

```
[kiyohaya@cw11 gbasf2KEK]$ echo $SHELL
/bin/bash
[kiyohaya@cw11 gbasf2KEK]$ source bashrc
```

csh,tcshと出た人は、
source cshrc
としてください。

```
[kiyohaya@cw11 gbasf2KEK]$ dirac-proxy-init -x
```

```
Generating proxy...
Enter Certificate password: █
Proxy generated:
subject      : /C=JP/O=KEK/OU=CRC/CN=HAYASAKA Kiyoshi/CN=proxy
issuer       : /C=JP/O=KEK/OU=CRC/CN=HAYASAKA Kiyoshi
identity     : /C=JP/O=KEK/OU=CRC/CN=HAYASAKA Kiyoshi
timeleft     : 23:59:59
rfc          : False
path         : /tmp/x509up_u21660
```

ここで聞かれているのは
パスフレーズです！

準備その3 (gbasf2の準備 2)

```
[kiyohaya@cw11 gbasf2KEK]$ dirac-configure defaults-Belle-KEK.cfg
Executing: /gpfs/home/belle/kiyohaya/tutorial20170624/gbasf2KEK/DIRAC/Core/scripts/dirac-configure.py defaults-Belle-KEK.cfg
Checking DIRAC installation at "/gpfs/home/belle/kiyohaya/tutorial20170624/gbasf2KEK"
Created vommdir file /gpfs/home/belle/kiyohaya/tutorial20170624/gbasf2KEK/etc/grid-security/vommdir/belle/voms.cc.kek.jp.lsc
Created vomses file /gpfs/home/belle/kiyohaya/tutorial20170624/gbasf2KEK/etc/grid-security/vomses/belle
```

```
[kiyohaya@cw11 gbasf2KEK]$ source BelleDIRAC/gbasf2/tools/setup
```

```
[kiyohaya@cw11 gbasf2KEK]$ gb2_proxy_init -g belle
```

```
Generating proxy...
```

```
Enter Certificate password: █
```

これもパスフレーズ

なんかごちゃごちゃ出ますが...

```
Proxy generated:
```

```
subject      : /C=JP/O=KEK/OU=CRC/CN=HAYASAKA Kiyoshi/CN=proxy/CN=proxy
issuer       : /C=JP/O=KEK/OU=CRC/CN=HAYASAKA Kiyoshi/CN=proxy
identity     : /C=JP/O=KEK/OU=CRC/CN=HAYASAKA Kiyoshi
timeleft     : 23: 53: 59
DIRAC group  : belle
rfc          : False
path        : /tmp/x509up_u21660
username    : kiyohaya
properties   : NormalUser
VOMS        : True
VOMS fqan   : ['/belle']
```

← これが残り時間。

この時間過ぎたら

もう1度 gb2_proxy_init してください

これで、gbasf2が使えるようになります。

残り時間の確認法

```
[kiyohaya@cw11 gbasf2KEK]$ voms-proxy-info -all
subject      : /C=JP/O=KEK/OU=CRC/CN=HAYASAKA Kiyoshi/CN=proxy/CN=proxy
issuer       : /C=JP/O=KEK/OU=CRC/CN=HAYASAKA Kiyoshi/CN=proxy
identity     : /C=JP/O=KEK/OU=CRC/CN=HAYASAKA Kiyoshi/CN=proxy
type         : proxy
strength     : 1024 bits
path         : /tmp/x509up_u21660
timeleft     : 23:51:33
key usage    : Digital Signature, Key Encipherment, Data Encipherment
=== V0 belle extension information ===
V0           : belle
subject      : /C=JP/O=KEK/OU=CRC/CN=HAYASAKA Kiyoshi
issuer       : /C=JP/O=KEK/OU=CRC/CN=host/voms.cc.kek.jp
attribute    : /belle/Role=NULL/Capability=NULL
attribute    : /belle/team/Role=NULL/Capability=NULL
timeleft     : 23:51:33
uri          : voms.cc.kek.jp:15020
```

← これが残り時間

gbasf2のセットアップ（2回目以降）

- 2回目以降は全部やる必要はありません。 ← gbasf2をinstallしたdirに移動

```
[kiyohaya@cw03 tutorial20170624]$ cd gbasf2KEK/
[kiyohaya@cw03 gbasf2KEK]$ source BelleDIRAC/gbasf2/tools/setup
[kiyohaya@cw03 gbasf2KEK]$ gb2_proxy_init -g belle
Generating proxy...
Enter Certificate password:
```

これで、gbasf2が使えるようになります。

ときどき、

```
gb2_update -V Belle-KEK
```

してください。これはgbasf2のupdate。日々進化してきますので、あなたのディレクトリにあるgbasf2は時代遅れになっているかもしれません。

(MC9中はupdateないかも...)

MC8の成果とその見方

- <https://confluence.desy.de/display/BI/Data+Production+MC8>

を見てください。

*** The LPN for all samples begins with '/belle/MC/release-00-08-00/DB00000208/MC8/'

*** For productions with both BGx1 and BGx0, the LPN is given for the BGx1 sample. The LPN for the BGx0 sample is identical except for the production ID.

Key	Scripts committed, ready for production	Jobs submitted	Released for analysis use	Not produced
-----	---	----------------	---------------------------	--------------

すんごい小さい字で、LPNは'/belle/MC/release-00-08-00/DB00000208/MC8/'で始まりますって書いてありますが、これ大事です。

Phase III - Y(4S) generic samples (1 ab-1)

Sample type	Number of events (10 ⁶)	Ratio without/with background	Production ID without/with background	LPN***	Link to json file	Estimated size (GB)
mixed	534.6	0.2/0.8	963/962	prod00000962/s00/e0000/4S/r00000/mixed	BGx0/BGx1	
charged	565.4	0.2/0.8	965/964	prod00000964/s00/e0000/4S/r00000/charged	BGx0/BGx1	

そしてここにもLPN

LPN/LFN

- Logical Path(File) Nameの略です。
- グリッド上の仮想的なディスクのpathだと思ってください。
- 実際のgrid SE(Storage Element)のpathとは異なりますが、通常意識しなくて大丈夫です。

gb2_ds_list

- まあ、lsです。

```
*** The LPN for all samples begins with '/belle/MC/release-00-08-00/DB00000208/MC8/'
```

uubar	00000967	BGx0	1.185786	sub00
2077, 2150, 2168	prod00000967/s00/e0000/4S/r00000/uubar/sub00			

- gb2_ds_list /belle/MC/release-00-08-00/DB00000208/MC8/prod00000966/s00/e0000/4S/r00000/uubar/sub01

アスタリスク（ワイルドカード）もいけます。

- gb2_ds_list -l /belle/MC/release-00-08-00/DB00000208/MC8/prod00000966/s00/e0000/4S/r00000/uubar/sub01/mdst_00376¥*_prod00000966_task000042¥*.root

-l を付けると... どこにファイルがあるか

グリッド上の
真の名前

```
[kiyohaya@cw12 example]$ gb2_ds_list -l /belle/MC/release-00-08-00/DB00000208/MC8/prod00000966/s00/e0000/4S/r00000/uubar/sub01/mdst_003760_prod00000966_task00004287.root
/belle/MC/release-00-08-00/DB00000208/MC8/prod00000966/s00/e0000/4S/r00000/uubar/sub01/mdst_003760_prod00000966_task00004287.root
2.0 GB
PNNL-TMP-SE - srm://se.hep.pnnl.gov:8443/srm/v2/server?SFN=/se/belle/TMP/belle/MC/release-00-08-00/DB00000208/MC8/prod00000966/s00/e0000/4S/r00000/uubar/sub01/mdst_003760_prod00000966_task00004287.root
/belle/MC/release-00-08-00/DB00000208/MC8/prod00000966/s00/e0000/4S/r00000/uubar/sub01/mdst_003761_prod00000966_task00004290.root
2.0 GB
PNNL-TMP-SE - srm://se.hep.pnnl.gov:8443/srm/v2/server?SFN=/se/belle/TMP/belle/MC/release-00-08-00/DB00000208/MC8/prod00000966/s00/e0000/4S/r00000/uubar/sub01/mdst_003761_prod00000966_task00004290.root
/belle/MC/release-00-08-00/DB00000208/MC8/prod00000966/s00/e0000/4S/r00000/uubar/sub01/mdst_003762_prod00000966_task00004291.root
2.0 GB
```

ファイルサイズ

詳細情報 (メタデータ)

```
gb2_ds_query_file -l /belle/MC/release-00-08-00/DB00000208/MC8/prod00000966/s00/e0000/4S/r00000/uubar/sub01/mdst_003769_prod00000966_task00004299.root
[kiyohaya@cw12 example]$ gb2_ds_query_file -l /belle/MC/release-00-08-00/DB00000208/MC8/prod00000966/s00/e0000/4S/r00000/uubar/sub01/mdst_003769_prod00000966_task00004299.root
mdst_003769_prod00000966_task00004299.root
  lfn: /belle/MC/release-00-08-00/DB00000208/MC8/prod00000966/s00/e0000/4S/r00000/uubar/sub01/mdst_003769_prod00000966_task00004299.root
  guid: AB817891-7D8C-5BA5-4C9E-478C0AAC7BAB
  date: 2017-03-11 05:53:37
  jobId: 46873326
  site: DIRAC.PNNL.us
  status: good
  nEvents: 296397
  experimentLow: 0
  experimentHigh: 0
  runLow: 0
  runHigh: 0
  eventLow: 131620002
  eventHigh: 660810001
  parentGuids:
```

詳細は

<https://confluence.desy.de/display/BI/Computing+Gbas+f2DataSetManagement>

ディレクトリ（？）構造

- /belle/MC (MC)
 - /belle/MC/MC作ったソフトウェアのバージョン
- /belle/data (data)
- /belle/user
 - /belle/user/dirac user名

ここにみなさんの
個人的なskim等が
入ります。

ちょっと話が行ったり来たりしますが、
考えた末ですので、
ご理解ください。

比較のために…

- もう1つ窓（端末）を開いて

gbasf2ではなく、**basf2**が動く環境を持っておきましょう。

別の窓で、**basf2用**の作業ディレクトリを用意してください。
そこで、`source /sw/belle2/tools/setup_belle2`して
`setuprel release-00-08-00`
してください。

```
[kiyohaya@cw08 tutorial20170624]$ mkdir basf_example
[kiyohaya@cw08 tutorial20170624]$ cd basf_example/
[kiyohaya@cw08 basf_example]$ source /sw/belle2/tools/setup_belle2
Belle II software tools set up at: /sw/belle2/tools
[kiyohaya@cw08 basf_example]$ setuprel release-00-08-00
Environment setup for release: release-00-08-00
Central release directory      : /cvmfs/belle.cern.ch/sl6/releases/release-00-08-00
```

ちょっと脱線：BASF2の便利なオプション

- `-i inputMdst`で指定された入力ファイルではなく、`-i` で指定されたファイルを読むようになります。steering fileを書き換えることなく、様々なmdstを読ませるのに便利。
 - `basf2 st.py -i sig.mdst.root`
 - `basf2 st.py -i bgA.mdst.root`
 - `basf2 st.py -i bgB.mdst.root`

一度固まったものは安易に書き換えないのが定石
- `-n` このオプションで指定されたevent数だけprocess する。
 - 今野さんの例では`process(analysis_main, 100)`としていたが、`process(analysis_main)`としておいて、`basf2 st.py -n 100`としても同じ。

作ったsteering fileはいきなり全イベントで試すのではなく、小数のイベントで動作を確認すべき。

ちょっと脱線：便利なEXAMPLE

- `ls /cvmfs/belle.cern.ch/sl6/releases/release-00-08-00/analysis/examples/tutorials/`

```
[kiyohaya@cw13 ~]$ ls /cvmfs/belle.cern.ch/sl6/releases/release-00-08-00/analysis/examples/tutorials/
B2A101-Y4SEventGeneration.dec          B2A303-MultipleDecays-Reconstruction.py  B2A601-ParticleStats.py
B2A101-Y4SEventGeneration.py           B2A304-B02RhoGamma-Reconstruction.py     B2A602-BestCandidateSelection.py
B2A102-ccbarEventGeneration.dec        B2A305-Btag+SingleMuon-Reconstruction.py B2A701-ContinuumSuppression_Input.py*
B2A102-ccbarEventGeneration.py         B2A306-B02RhoGamma-withPi0Veto.py        B2A702-ContinuumSuppression_MVATrain.py*
B2A103-SimulateAndReconstruct-withoutBeamBkg.py B2A401-KFit-MassFit.py                  B2A703-ContinuumSuppression_MVAExpert.py*
B2A104-SimulateAndReconstruct-withBeamBkg.py B2A403-KFit-VertexFit.py                B2A801-FlavorTagger.py
B2A201-LoadMCParticles.py              B2A404-Rave-VertexFit.py                B2A802-FlavorTagger-BelleMC.py
B2A202-LoadReconstructedParticles.py    B2A405-Rave-MassVertexFit.py            B2A901-PrintOutDataStore.py
B2A301-Dstar2D0Pi-Reconstruction.py    B2A406-Rave-DecayStringVertexFit.py     B2A902-PrintOutMCParticles.py
B2A302-B02D0Pi0-D02Pi0Pi0-Reconstruction.py B2A410-TagVertex.py                     B2A903-PrintOutVariableValues.py
```

過去に行われたbasf2 tutorialのexample.ファイル名で
だいたいなにやってるのかわかるとおもいます。

release-00-08-00の部分は自分が使いたいバージョンに
書き換えてください。

ちょっと脱線：普通のEXAMPLE

- ls /cvmfs/belle.cern.ch/sl6/releases/release-00-08-00/analysis/examples/

```
[kiyohaya@cw13 ~]$ ls /cvmfs/belle.cern.ch/sl6/releases/release-00-08-00/analysis/examples/
BtagBsigReconstruction.py  deepLearning/          flavorTaggerEfficiency.py*  printMCParticleArray.py  variablesToHistogram.py
FEI/                       exampleEvtgenDecayFiles/  ntuplemaker.py*           standardPi0.py
art/                       expert_per_event.py     printDataStore.py         tutorials/
```

analysisのところは知りたいmodule名を入れる

関係ありそうなのはgenerator

```
[kiyohaya@cw13 ~]$ ls /cvmfs/belle.cern.ch/sl6/releases/release-00-08-00/generators/examples/
AafhGeneration.py          BhWide_10events.txt    FragmentationOnly.py      KKM CandFragmentation.py  SADreaderInput.root
AafhGenerationWithPreselection.py  CRYGenerationOnly.py  HepEvtReaderFull.py      Kora1WPlots.py           TeeggGenerationOnly.py
BBBremGenerationOnly.py      ContinuumGenOnly.py   HepEvtReaderMaster.py    ParticleGunFull.py       VertexGunFull.py
BHWideGenerationOnly.py      Cosmics.py            KKGenGenerationMuMuOnly.py  ParticleGunPlots.py      cry.setup
BHWidePlots.py              EvtGenFullSim.py     KKGenGenerationOnly.py    PhokharaGenerationOnly.py
BabayagaNLOGenerationOnly.py  EvtGenGenerationOnly.py  KKMCF ForFragmentation.py  RootToHepEvt.py
BeamParameters.py           EvtGenUpsilon1S.py    KKMCF ForFragmentationKEKB.py  SADreader.py
```

いろんなsampleをgenerationするsteering fileがある。

これとB2A103-SimulateAndReconstruct-withoutBeamBkg.py

を組み合わせると検出器シミュレーション入りのmdstが作れる。withはBGx1

basf2でMC generation

basf2の方の窓で！

```
cp /cvmfs/belle.cern.ch/sl6/releases/release-00-08-00/generators/examples/EvtGenGenerationOnly.py .
```

```
[kiyohaya@cw08 basf_example]$ cp /cvmfs/belle.cern.ch/sl6/releases/release-00-08-00/generators/examples/EvtGenGenerationOnly.py .
[kiyohaya@cw08 basf_example]$ basf2 EvtGenGenerationOnly.py
[INFO] Steering file: EvtGenGenerationOnly.py
[RESULT] Starting event processing, random seed is set to 'c5b713d5a5f7d8605c6ba190b5cb62505ea441b1ae0d4d79182c3ffa42e5fe6'
[WARNING] Failed to get dbstore/BeamParameters from local database /gpfs/home/belle/kiyohaya/tutorial20170624/basf_example/localdb/database.txt. { module: -global- }
[INFO] Found more than one payload for the following keys: ECLDigitEnergyConstantsHigh, ECLDigitTimeConstants, EKLMDigitizationParameters, EKLMSimulationParameters, EKLMTIMECalibration, FEI_Belle2_Generic_2016_Track12_1_B+: 00557c85b99c2929e75a53bf385d0243662f9048_0608c062be63dcdacc4e36a82f4cdab1ca1015a1, FEI_Belle2_Generic_2016_Track12_1_B+: 0aee1089e69139561108c644fa8079e735ec7127_5676cc5fa259cc83276e04c557017ed0ad3761b1, FEI_Belle2_Generic_2016_Track12_1_B+: 0efaf32cd51853f6c313a06bde4df1b3c1f9f742_db51b913ebe3517b294644cb56ba207d96b681c3, FEI_Belle2_Generic_2016_Track12_1_B+: 11e4e1fe6e44d7b70e8abbd50f24b50ccale3cb_2f0a9ef5d064b4e5456b0016740354c249788100, FEI_Belle2_Generic_2016_Track12_1_B+: 135dce90f4a69a77c83ff42348a8e0aa81f8a6ee_012250c9b56cb23eb35e4579ba9516f2f10fac9a, FEI_Belle2_Generic_2016_Track12_1_B+: 1964df5afb1b75029f6cae654d91775ee6ccc09e_3990f9fefba6b26d5fe39d6dc1ca7c002130600, FEI_Belle2_Generic_2016_Track12_1_B+: 1a3a3213270ad2e0b433a73d78aeb4b177e072d4_e3a92be16d0c832c729a08855f405ffeaf76ad46, FEI_Belle2_Generic_2016_Track12_1_B+: 1f8cbdd4699b6696f2393d417c4c39a95ac9bbf6_8281032a3fed235b43f735b6f14406474e43951
```

• $B^0\bar{B}^0$ を100 event生成する。

```
[kiyohaya@cw14 basf_example]$ ls
Belle2FileCatalog.xml EvtGenGenerationOnly.py evtgen_upsilon4s.root
```

gbasf2でMC generation

- gbasf2の方の窓で、新しく gbasf2作業用のディレクトリを作って、(今いるところは、gbasf2の諸々が入ったディレクトリなので、そことは独立な方がいい)

新しく作ったディレクトリで

```
cp /cvmfs/belle.cern.ch/sl6/releases/release-00-08-00/generators/examples/EvtGenGenerationOnly.py .
```

早速、「gbasf2 EvtGenGenerationOnly.py」としたいところだが、そうは問屋が卸さない。

SOFTWARE VERSIONの指定

- basf2の場合はsetuprelで使うソフトウェアのバージョン (release)を指定した。
- gbasf2では、2つの方法で、releaseを指定できる。
 - gbasf2の-sオプションで
 - `gbasf2 -s release-00-08-00 ...`
とすると、release-00-08-00を使ってジョブは実行される
 - Steering fileの中に書く
 - Steering fileの中に
`gb2_setuprel = 'release-00-08-00'`
と書くと、release-00-08-00を使ってジョブは実行される

PROJECTの指定

- 今の段階ではピンとこないと思います。
- グリッドジョブは、複数同時に実行されるので、プロジェクトという塊を作ります。
- プロジェクト名は、かぶるとややこしいので、自分の名前 + 作業の名前 + 日付くらいにしておくといいです。(31文字まで)

- 成果物は

/belle/user/自分のdirac名/プロジェクト名

に格納されます。

何をしたときのプロジェクト名なのかをちゃんと自分のログに記録しておくことを強く勧めます。

gbasf2のジョブ投入

```
[kiyohaya@cw12 example]$ gbasf2 EvtGenGenerationOnly.py -p hayasaka_gentestu4s onejob -s release-00-08-00
```

```
*****
```

```
***** Project summary *****
```

```
** Project name: hayasaka_gentestu4s_onejob  
** Dataset path: /belle/user/kiyohaya/hayasaka_gentestu4s_onejob  
** Steering file: EvtGenGenerationOnly.py  
** Job owner: kiyohaya @ belle (21:11:55)  
** Preferred site / SE: None / None  
** Input files: ['None']  
** Processed events: 0 events  
** Estimated CPU time per job: 0 min
```

```
*****
```

```
Are you sure to submit the project?  
Please enter Y or N: █
```

gbasf2のジョブ投入

```
[kiyohaya@cw12 example]$ gbasf2 EvtGenGenerationOnly.py -p hayasaka_gentestu4s onejob -s release-00-08-00
```

```
*****
```

```
***** Project summary *****
```

```
** Project name: hayasaka_gentestu4s_onejob
```

```
** Dataset path: /belle/user/kiyohaya/hayasaka_gentestu4s_onejob
```

```
** Steering file: EvtGenGenerationOnly.py
```

```
Please enter Y or N: y
```

```
Initialize metadata for the project:
```

```
No attribute. Initialize Dataset...
```

```
Dataset initialization: OK
```

```
Dataset metadata attributes already exist (30): OK
```


```
Successfully finished.
```

```
JobID = 52277114
```

今回投入されたジョブのジョブ番号

ジョブの状態の確認

ジョブ番号を
使った確認



```
[kiyohaya@cw12 example]$ gb2_job_status -j 52277114
```

Job id	Status	MinorStatus	ApplicationStatus	Site
52277114	Done	Execution Complete	Done	LCG.KMI.jp

```
--- Summary of Selected Jobs ---
```

```
Completed: 0 Deleted: 0 Done: 1 Failed: 0 Killed: 0 Running: 0 Stalled: 0  
Waiting: 0
```

```
[kiyohaya@cw12 example]$ gb2_job_status -p hayasaka_gentestu4s_onejob
```

```
1 jobs are selected.
```

Job id	Status	MinorStatus	ApplicationStatus	Site
52277114	Done	Execution Complete	Done	LCG.KMI.jp

```
--- Summary of Selected Jobs ---
```

```
Completed: 0 Deleted: 0 Done: 1 Failed: 0 Killed: 0 Running: 0 Stalled: 0  
Waiting: 0
```

<https://confluence.desy.de/display/BI/Computing+Gbasf2JobManagement>

Statusについて

- Done →めでたく終了
- Failed →失敗
- Completed →basf2の実行は終わったがファイル転送がまだ
Completedは、待つ
Failedは、ローカル(basf2)で実行してみて本当に問題ないか確認

(画面への) 出力結果の取得

```
[kiyohaya@cw10 example]$ gb2_job_output -j 52154646  
1 jobs are selected.  
Please wait...
```

自分のジョブの
ジョブ番号

```
Result for jobs: ['52154646']
```

```
=====  
Downloaded: "Job output sandbox retrieved in /gpfs/home/belle/kiyohaya/tutorial2  
0170624/example/52154646"
```

ジョブ番号のディレクトリが新しく
できている

```
[kiyohaya@cw10 example]$ ls  
52154646/ EvtGenGenerationOnly.py
```

```
[kiyohaya@cw10 example]$ ls 52154646/  
Script1_basf2helper.py.log evtgen_upsilon4s_1.metadata job.info std.out
```

basfの出したログ

gbasfの出したログ

Script1_basf2helper.py.log の中を見てみると

```
[INFO] 3 511 (B0) E: 5.496e+00 m: 5.280e+00 p: ( 5.954e-01, 1.552e-02, 1.407e+00) v: (-4.107e-04, -2.619e-06, 1.039e-02), t: 0.000e+00, 5.455e-04, s: 1, c: 0
[INFO] 10 311 (K0) E: 5.542e-01 m: 4.976e-01 p: ( 1.723e-01, -6.881e-02, -1.584e-01) v: ( 1.361e-03, 4.354e-05, 1.458e-02), t: 5.455e-04, 5.455e-04, s: 1, c: 0
[INFO] 27 310 (K S0) E: 5.542e-01 m: 4.976e-01 p: ( 1.723e-01, -6.881e-02, -1.584e-01) v: ( 1.361e-03, 4.354e-05, 1.458e-02), t: 5.455e-04, 1.097e-01, s: 1, c: 0
[INFO] 45 211 (pi+) E: 3.151e-01 m: 1.396e-01 p: ( 1.635e-01, 1.215e-01, -1.957e-01) v: ( 1.019e+00, -4.064e-01, -9.213e-01), t: 1.097e-01, inf, s: 3, c: 1
[INFO] 46 -211 (pi-) E: 2.391e-01 m: 1.396e-01 p: ( 8.874e-03, -1.903e-01, 3.731e-02) v: ( 1.019e+00, -4.064e-01, -9.213e-01), t: 1.097e-01, inf, s: 3, c: -1
[INFO] 11 113 (rho0) E: 9.240e-01 m: 7.849e-01 p: (-4.393e-01, 1.700e-01, 1.260e-01) v: ( 1.361e-03, 4.354e-05, 1.458e-02), t: 5.455e-04, 5.455e-04, s: 1, c: 0
[INFO] 28 211 (pi+) E: 5.321e-01 m: 1.396e-01 p: (-4.815e-01, -1.775e-01, 1.822e-02) v: ( 1.361e-03, 4.354e-05, 1.458e-02), t: 5.455e-04, inf, s: 3, c: 1
[INFO] 29 -211 (pi-) E: 3.919e-01 m: 1.396e-01 p: ( 4.221e-02, 3.475e-01, 1.077e-01) v: ( 1.361e-03, 4.354e-05, 1.458e-02), t: 5.455e-04, inf, s: 3, c: -1
[INFO] 12 -421 (anti-D0) E: 2.066e+00 m: 1.865e+00 p: (-2.094e-02, 5.280e-01, 7.147e-01) v: ( 1.361e-03, 4.354e-05, 1.458e-02), t: 5.455e-04, 7.219e-04, s: 1, c: 0
[INFO] 30 213 (rho+) E: 9.790e-01 m: 7.416e-01 p: ( 4.206e-01, 1.091e-01, 4.685e-01) v: ( 1.307e-03, 1.395e-03, 1.641e-02), t: 7.219e-04, 7.219e-04, s: 1, c: 1
[INFO] 47 211 (pi+) E: 5.091e-01 m: 1.396e-01 p: ( 5.329e-03, 2.715e-01, 4.073e-01) v: ( 1.307e-03, 1.395e-03, 1.641e-02), t: 7.219e-04, inf, s: 3, c: 1
[INFO] 48 111 (pi0) E: 4.699e-01 m: 1.350e-01 p: ( 4.153e-01, -1.624e-01, 6.124e-02) v: ( 1.307e-03, 1.395e-03, 1.641e-02), t: 7.219e-04, 7.219e-04, s: 1, c: 0
[INFO] 51 22 (gamma) E: 2.565e-01 m: 0.000e+00 p: ( 2.494e-01, -5.608e-02, -2.080e-02) v: ( 1.307e-03, 1.395e-03, 1.641e-02), t: 7.219e-04, inf, s: 3, c: 0
[INFO] 52 22 (gamma) E: 2.135e-01 m: 0.000e+00 p: ( 1.659e-01, -1.063e-01, 8.204e-02) v: ( 1.307e-03, 1.395e-03, 1.641e-02), t: 7.219e-04, inf, s: 3, c: 0
[INFO] 31 13 (mu-) E: 5.867e-01 m: 1.057e-01 p: (-1.051e-01, 1.880e-01, 5.354e-01) v: ( 1.307e-03, 1.395e-03, 1.641e-02), t: 7.219e-04, inf, s: 3, c: -1
[INFO] 32 -14 (anti-nu_mu) E: 5.001e-01 m: 1.490e-08 p: (-3.365e-01, 2.308e-01, -2.892e-01) v: ( 1.307e-03, 1.395e-03, 1.641e-02), t: 7.219e-04, inf, s: 3, c: 0
[INFO] 33 22 (gamma) E: 1.458e-05 m: 1.563e-13 p: ( 9.874e-06, 8.953e-06, 5.897e-06) v: ( 1.307e-03, 1.395e-03, 1.641e-02), t: 7.219e-04, inf, s: 259, c: 1
[INFO] 13 223 (omega) E: 1.672e+00 m: 7.845e-01 p: ( 1.066e+00, -5.362e-01, 8.700e-01) v: ( 1.361e-03, 4.354e-05, 1.458e-02), t: 5.455e-04, 5.455e-04, s: 1, c: 0
[INFO] 34 -211 (pi-) E: 7.870e-01 m: 1.396e-01 p: ( 6.037e-01, -3.158e-01, 3.683e-01) v: ( 1.361e-03, 4.354e-05, 1.458e-02), t: 5.455e-04, inf, s: 3, c: -1
[INFO] 35 211 (pi+) E: 2.857e-01 m: 1.396e-01 p: (-3.161e-02, -1.476e-01, 1.985e-01) v: ( 1.361e-03, 4.354e-05, 1.458e-02), t: 5.455e-04, inf, s: 3, c: 1
[INFO] 36 111 (pi0) E: 5.993e-01 m: 1.350e-01 p: ( 4.937e-01, -7.283e-02, 3.032e-01) v: ( 1.361e-03, 4.354e-05, 1.458e-02), t: 5.455e-04, 5.455e-04, s: 1, c: 0
[INFO] 49 22 (gamma) E: 1.775e-02 m: 8.395e-10 p: (-1.766e-03, -1.725e-02, 3.766e-03) v: ( 1.361e-03, 4.354e-05, 1.458e-02), t: 5.455e-04, inf, s: 3, c: 0
[INFO] 50 22 (gamma) E: 5.816e-01 m: 8.395e-10 p: ( 4.955e-01, -5.558e-02, 2.994e-01) v: ( 1.361e-03, 4.354e-05, 1.458e-02), t: 5.455e-04, inf, s: 3, c: 0
[INFO] 14 111 (pi0) E: 2.802e-01 m: 1.350e-01 p: (-1.825e-01, -7.749e-02, -1.448e-01) v: ( 1.361e-03, 4.354e-05, 1.458e-02), t: 5.455e-04, 5.455e-04, s: 1, c: 0
[INFO] 37 22 (gamma) E: 1.811e-01 m: 8.550e-10 p: (-1.117e-01, -1.139e-01, -8.593e-02) v: ( 1.361e-03, 4.354e-05, 1.458e-02), t: 5.455e-04, inf, s: 3, c: 0
[INFO] 38 22 (gamma) E: 9.904e-02 m: 8.550e-10 p: (-7.086e-02, 3.636e-02, -5.887e-02) v: ( 1.361e-03, 4.354e-05, 1.458e-02), t: 5.455e-04, inf, s: 3, c: 0
[INFO] Creating file catalog /home/dirac/exec/DIRAC_fwzJKppilot/52154646/Belle2FileCatalog.xml
[INFO] Write TTree tree
[INFO] Write TTree persistent
[INFO] =====
[INFO] Error summary: 0 errors and 1 warnings occurred.
[WARNING] Failed to get dbstore/BeamParameters from local database /home/dirac/exec/DIRAC_fwzJKppilot/52154646/localdb/database.txt.
[INFO] =====
```

Name	Calls	VMemory(MB)	Time(s)	Time(ms)/Call
EventInfoSetter	101	0	0.00	0.00 +- 0.00
Progress	100	0	0.00	0.01 +- 0.01
Gearbox	100	0	0.00	0.00 +- 0.00
EvtGenInput	100	0	3.78	37.80 +- 249.11
RootOutput	100	0	0.00	0.05 +- 0.06
PrintMCParticles	100	0	4.75	47.54 +- 455.87
Total	101	0	8.54	84.58 +- 513.83

確かにbasf2の時の出力と同じようなものが出ている

MCファイルは？

プロジェクト名を
指定

```
[kiyohaya@cw05 gbasf2KEK]$ gb2_ds_list hayasaka_gentestu4s_onejob  
/belle/user/kiyohaya/hayasaka_gentestu4s_onejob/evtgen_upsilon4s_1.root
```

basf2の結果と比べると...

```
[kiyohaya@cw14 basf_example]$ ls  
Belle2FileCatalog.xml  EvtGenGenerationOnly.py  evtgen_upsilon4s.root
```

_1が付いている？！

この理由は後でわかります。

が、一旦別の話に...

GRID上にあるのはわかったけど...

- これでは画餅ですね。

プロジェクト名

```
[kiyohaya@cw05 example]$ gb2_ds_get hayasaka_gentestu4s_onejob
Files to download to /gpfs/home/belle/kiyohaya/tutorial20170624/example/hayasaka_gentestu4s_onejob :
/belle/user/kiyohaya/hayasaka_gentestu4s_onejob/evtgen_upsilon4s_1.root
Do you want to download files:
Please type [Y] or [N]: y

Download 1 files from SE
__getFile: Using 1 streams
__getFile: Executing transfer of srm://nrmfe01.hepl.phys.nagoya-u.ac.jp:8444/srm/managerv2?SFN=/belle/TMP/belle/user/kiyohaya/hayasaka_gentestu4s_onejob/evtgen_upsilon4s_1.root to file:/gpfs/home/belle/kiyohaya/tutorial20170624/example/hayasaka_gentestu4s_onejob/evtgen_upsilon4s_1.root

Successfully downloaded files:
/belle/user/kiyohaya/hayasaka_gentestu4s_onejob/evtgen_upsilon4s_1.root in /gpfs/home/belle/kiyohaya/tutorial20170624/example/hayasaka_gentestu4s_onejob
```

プロジェクト名のdirが

```
[kiyohaya@cw05 example]$ ls hayasaka_gentestu4s_onejob/
evtgen_upsilon4s_1.root
```

できてそこに
ファイルが配置

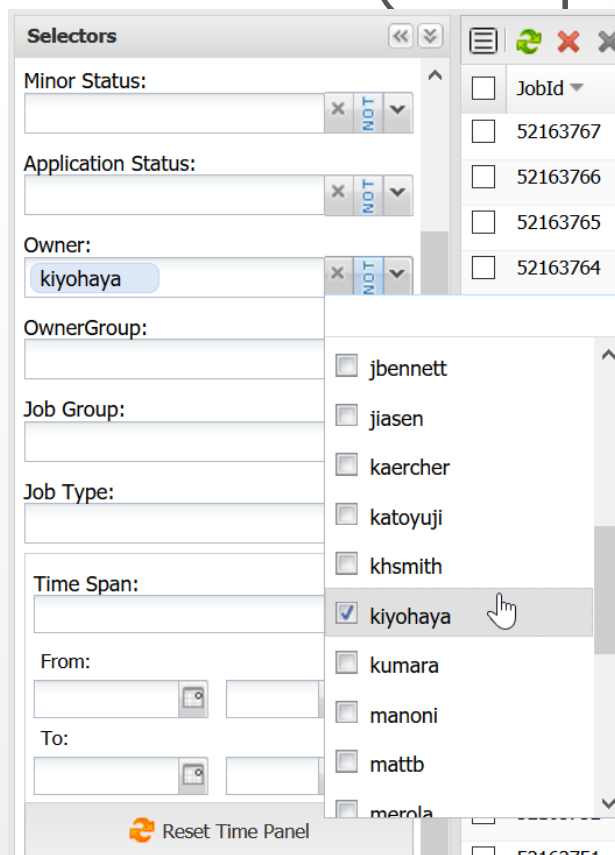
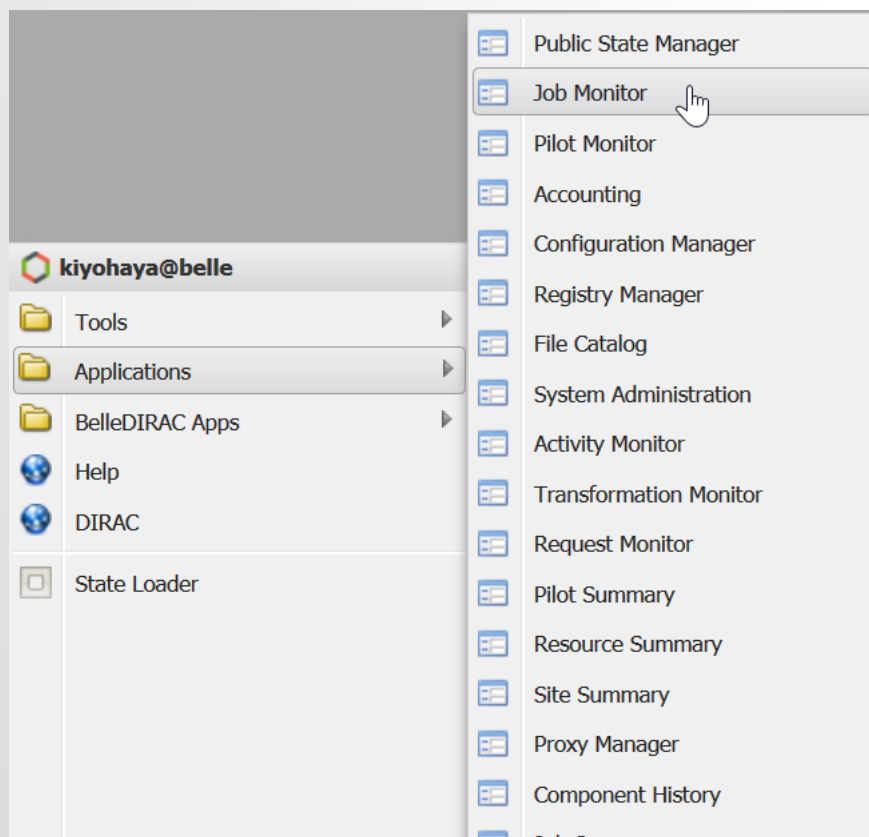
実際は

- ユーザーが実際にダウンロードしてくるのは解析用のroot file
- mdstなどのイベントが入ってるファイルはグリッド上のジョブで処理されるため、ダウンロードしてくる必要はないはず。

gb2_ds_get でMCサンプルを大量にダウンロードするのは自粛してください。

自分のJobの確認の仕方 (Web編)

- <https://dirac.cc.kek.jp:8443> にアクセス
(Web portal)



ownerに
自分の名前
を入れて
おく。→
submit

Job Monitor

Selectors << >>

Application Status: [] x NOT v ^

Owner: [kiyohaya] x NOT v

OwnerGroup: [] x NOT v

Job Group: [] x NOT v

Job Type: [] x NOT v

Time Span: [] v

From: [] [] v

To: [] [] v

Reset Time Panel

Submit Reset Refresh

自分のJobの確認の仕方 (Web編)

- Submitを押す



- 情報の更新は  を押す。

<input type="checkbox"/> JobId v	Status	MinorS	ApplicationStatus	Site	JobNar	LastUpdate[UTC]	LastSignOfLife[UTC]	SubmissionTime[UTC]
<input type="checkbox"/> 52277114	■ Matched	Inst...	Unknown	LCG.KMI.jp	None	2017-06-23 03:34:50	2017-06-23 03:34:50	2017-06-23 03:34:50
<input type="checkbox"/> JobId v	Status	MinorS	ApplicationStatus	Site	JobNar	LastUpdate[UTC]	LastSignOfLife[UTC]	SubmissionTime[UTC]
<input type="checkbox"/> 52277114	■ Running	App...	Done	LCG.KMI.jp	None	2017-06-23 03:35:41	2017-06-23 03:35:45	2017-06-23 03:34:50

JobId	Status	MinorStatus	ApplicationStatus
52277114		Executio...	Done
52275174			
52274949			
52274948	Waiting		
52274568	Killed		
52274567	Killed		
52273017	Completed		
52154646	Done		

WEB PORTALでのPROJECTの確認

- JobIDとかStatusとかの上をマウスカーソル動かすと、|▼|が出てくるのでクリック。
- メニューが出てくるのでColumnsをクリック
- JobGroupにチェックを入れる

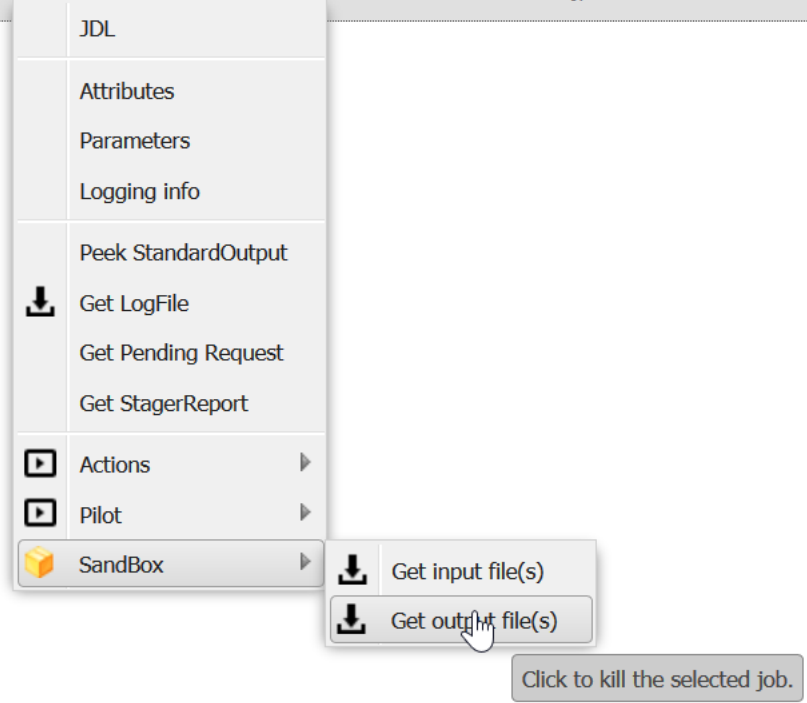
ここにプロジェクトが出る

JobId	Status	MinorStatus	ApplicationStatus	Site	JobName	LastUpdate[UTC]	LastSignOfLife[UTC]	SubmissionTime[UTC]	JobGroup	Owner
52277114	Done	Executio...	Done	LCG.KMI.jp	None	2017-06-23 03:35:46	2017-06-23 03:35:46	2017-06-23 03:34:50	hayasaka_gentestu4s_onejob	kiyohaya

WEB PORTALから出力を得る

<input type="checkbox"/>	JobId ▾	Status	MinorStatus	ApplicationStatus	Site	JobNar	LastUpdate[UTC]	LastSign
<input type="checkbox"/>	52154646	Done	Execution Complete	Done	SSH.KMI.jp	None	2017-06-20 01:19:14	2017-06-

<input type="checkbox"/>	JobId ▾	Status	MinorStatus	ApplicationStatus	Site	JobNar	LastU
<input type="checkbox"/>	52154646	Done	Execution Complete	Done	SSH.KMI.jp	None	2017-



- JDL
- Attributes
- Parameters
- Logging info
- Peek StandardOutput
- ↓ Get LogFile
- Get Pending Request
- Get StagerReport
- ▶ Actions
- ▶ Pilot
- ▶ SandBox
 - ↓ Get input file(s)
 - ↓ Get output file(s)

Click to kill the selected job.

右クリック (Mac?)
Sandbox→
Get output file(s)

但し、tar.gzが来る
ので、tar.gzを解凍
できるソフトが
入っている必要が
あります。

<http://www7a.biglobe.ne.jp/~schezo/>
早坂はLhaplusを使っています。

投げたジョブをなかったことにしたいときは…

<input type="checkbox"/>	JobId ▾	Status	MinorStatus	ApplicationSt	Site	JobName	LastUpdate[UTC]
<input type="checkbox"/>	52312086	Running	Application	Running	LCG.Frascati.it	mdst_00...	2017-06-23 15:24:51
<input type="checkbox"/>	52312085	Running	Application	Running	LCG.Cosenza.it	mdst_00...	2017-06-23 15:30:26
<input type="checkbox"/>	52312084	Running	Application	Running	LCG.Cosenza.it	mdst_00...	2017-06-23 15:33:44
<input type="checkbox"/>	52312083	Running	Application	Running	LCG.Cosenza.it	mdst_00...	2017-06-23 15:42:23
<input type="checkbox"/>	52312082	Running	Application	Running	LCG.Frascati.it	mdst_00...	2017-06-23 15:23:45
<input type="checkbox"/>	52312081	Running	Application	Running	LCG.Cosenza.it	mdst_00...	2017-06-23 15:43:41

- チェックを入れて…

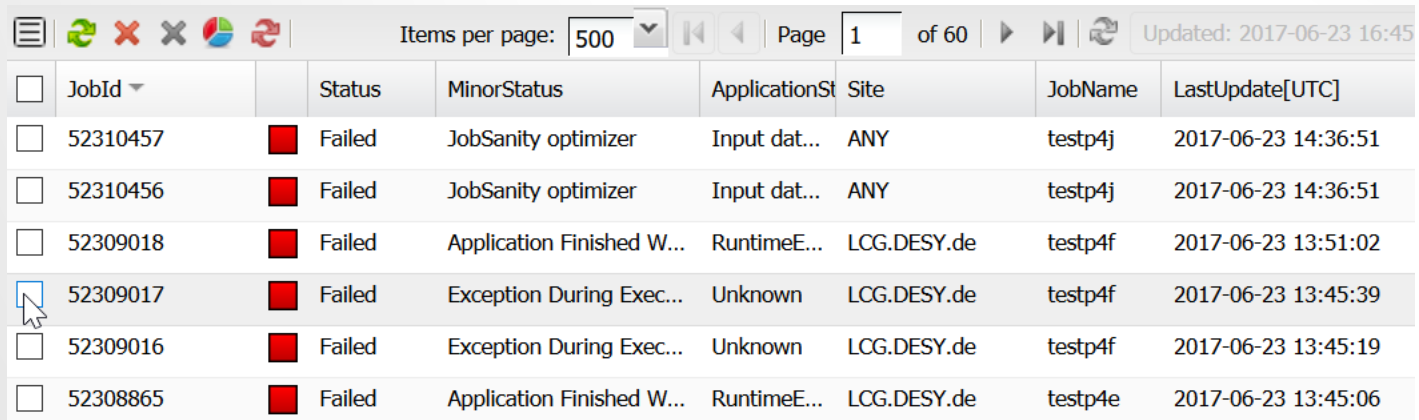
<input type="checkbox"/>	JobId	Status	MinorStatus	ApplicationSt	Site	JobName	LastUpdate[UTC]
<input checked="" type="checkbox"/>	52312086	Running	Application	Running	LCG.Frascati.it	mdst_00...	2017-06-23 15:24:51
<input checked="" type="checkbox"/>	52312085	Running	Application	Running	LCG.Cosenza.it	mdst_00...	2017-06-23 15:30:26
<input checked="" type="checkbox"/>	52312084	Running	Application	Running	LCG.Cosenza.it	mdst_00...	2017-06-23 15:33:44
<input type="checkbox"/>	52312083	Running	Application	Running	LCG.Cosenza.it	mdst_00...	2017-06-23 15:42:23
<input type="checkbox"/>	52312082	Running	Application	Running	LCG.Frascati.it	mdst_00...	2017-06-23 15:23:45

Items per page: 500 Page 1 of 1

Kill

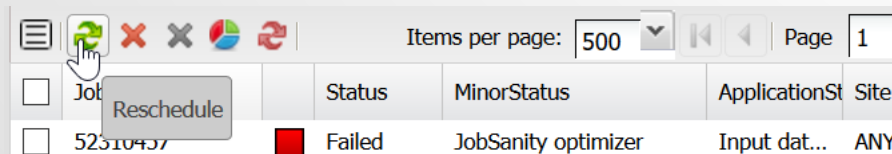
失敗したジョブをやり直したい時は

- もう絶対、ゼッタイ自信があって、「Failしたのはおらのせいじゃねー、サイトのせーだー」という人は、



<input type="checkbox"/>	JobId	Status	MinorStatus	ApplicationSt	Site	JobName	LastUpdate[UTC]
<input type="checkbox"/>	52310457	Failed	JobSanity optimizer	Input dat...	ANY	testp4j	2017-06-23 14:36:51
<input type="checkbox"/>	52310456	Failed	JobSanity optimizer	Input dat...	ANY	testp4j	2017-06-23 14:36:51
<input type="checkbox"/>	52309018	Failed	Application Finished W...	RuntimeE...	LCG.DESY.de	testp4f	2017-06-23 13:51:02
<input checked="" type="checkbox"/>	52309017	Failed	Exception During Exec...	Unknown	LCG.DESY.de	testp4f	2017-06-23 13:45:39
<input type="checkbox"/>	52309016	Failed	Exception During Exec...	Unknown	LCG.DESY.de	testp4f	2017-06-23 13:45:19
<input type="checkbox"/>	52308865	Failed	Application Finished W...	RuntimeE...	LCG.DESY.de	testp4e	2017-06-23 13:45:06

チェック入れたら...



<input checked="" type="checkbox"/>	Job	Status	MinorStatus	ApplicationSt	Site
<input checked="" type="checkbox"/>	52310457	Failed	JobSanity optimizer	Input dat...	ANY

コマンドラインでプロジェクト丸ごと可能

- `gb2_job_kill -p プロジェクト名`
- `gb2_job_reschedule -p プロジェクト名`

などなど

<https://confluence.desy.de/display/BI/Computing+Gbasf2JobManagement>
にいろいろ載ってます

MC大量生成

さっきと違う
プロジェクト名

- -rを使う

```
[kiyohaya@cw05 example]$ gbasf2 EvtGenGenerationOnly.py -r 10 -p hayasaka_gentestu4s_tenjob  
-s release-00-08-00  
*****  
***** Project summary *****  
** Project name: hayasaka_gentestu4s_tenjob  
** Dataset path: /belle/user/kiyohaya/hayasaka_gentestu4s_tenjob  
** Steering file: EvtGenGenerationOnly.py  
** Job owner: kiyohaya @ belle (23:40:59)  
** Preferred site / SE: None / None  
** Input files: ['None']  
** Processed events: 0 events * 10 jobs  
** Estimated CPU time per job: 0 min  
*****  
Are you sure to submit the project?  
Please enter Y or N: █
```

10個ジョブ投げる

ということなのか？

- MC sampleを生成する時、1つのジョブでなが〜く、大きなファイルを作るのは現実的ではありません。
- -rは全く同じ、でもrandom seedは違う、MC 生成ジョブを指定した数だけ (max 100)走らせてくれます。
- Random seedは勝手に設定されます。(これは、1つの時も同じ。十分に長い乱数なので、適当に設定したrandom seedではまず、乱数が重なることがないだろう、ということです。ここ、ソフトウェアグループ立ち上げ時に喧々囂々の議論がありました。TRandom3だったはず)

MC大量生成

さっきと違う
プロジェクト名

- -rを使う

```
[kiyohaya@cw05 example]$ gbasf2 EvtGenGenerationOnly.py -r 10 -p hayasaka_gentestu4s_tenjob  
-s release-00-08-00  
*****  
***** Project summary *****  
** Project name: hayasaka_gentestu4s_tenjob  
** Dataset path: /belle/user/kiyohaya/hayasaka_gentestu4s_tenjob  
** Steering file: EvtGenGenerationOnly.py  
** Job owner: kiyohaya @ belle (23:40:59)  
** Preferred site / SE: None / None  
** Input files: ['None']  
** Processed events: 0 events * 10 jobs  
** Estimated CPU time per job: 0 min  
*****  
Are you sure to submit the project?  
Please enter Y or N:   
Please enter Y or N: y  
Initialize metadata for the project:  
No attribute. Initialize Dataset...  
Dataset initialization: OK  
Dataset metadata attributes already exist (30): OK  
Successfully finished.  
JobID = 52280607 ... 52280616 (10 jobs)
```

Job番号
10個

プロジェクト名が便利になってくる

```
[kiyohaya@cw05 example]$ gb2_job_status -p hayasaka_gentestu4s_tenjob  
10 jobs are selected.
```

Job id	Status	MinorStatus	ApplicationStatus	Site
52280607	Running	Job Initialization	Unknown	ARC. KIT. de
52280608	Running	Application	Uploading	LCG. KEK. jp
52280609	Running	Application	Unknown	LCG. HEPHY. at
52280610	Running	Application	Unknown	LCG. HEPHY. at
52280611	Running	Application	Unknown	LCG. HEPHY. at
52280612	Running	Application	Running	LCG. KMI. jp
52280613	Matched	Submitted To CE	Unknown	LCG. ULAKBIM. tr
52280614	Running	Job Initialization	Unknown	LCG. HEPHY. at
52280615	Running	Job Initialization	Unknown	LCG. HEPHY. at
52280616	Running	Job Initialization	Unknown	LCG. HEPHY. at

できあがりは？

```
[kiyohaya@cw05 example]$ gb2_ds_list hayasaka_gentestu4s_tenjob  
/belle/user/kiyohaya/hayasaka_gentestu4s_tenjob/evtgen_upsilon4s_0.root  
/belle/user/kiyohaya/hayasaka_gentestu4s_tenjob/evtgen_upsilon4s_1.root  
/belle/user/kiyohaya/hayasaka_gentestu4s_tenjob/evtgen_upsilon4s_2.root  
/belle/user/kiyohaya/hayasaka_gentestu4s_tenjob/evtgen_upsilon4s_3.root  
/belle/user/kiyohaya/hayasaka_gentestu4s_tenjob/evtgen_upsilon4s_4.root  
/belle/user/kiyohaya/hayasaka_gentestu4s_tenjob/evtgen_upsilon4s_5.root  
/belle/user/kiyohaya/hayasaka_gentestu4s_tenjob/evtgen_upsilon4s_6.root  
/belle/user/kiyohaya/hayasaka_gentestu4s_tenjob/evtgen_upsilon4s_7.root  
/belle/user/kiyohaya/hayasaka_gentestu4s_tenjob/evtgen_upsilon4s_8.root  
/belle/user/kiyohaya/hayasaka_gentestu4s_tenjob/evtgen_upsilon4s_9.root
```

番号ついてる
意味がわかり
ましたか？

-rで出来上がったファイルの重複を避けるため
出来上がったファイル名の後ろに番号を付けます

メタデータは？

```
[kiyohaya@cw05 example]$ gb2_ds_query_file /belle/user/kiyohaya/hayasaka_gentestu4s_tenjob/  
evtgen_upsilon4s_0.root  
evtgen_upsilon4s_0.root | lfn: /belle/user/kiyohaya/hayasaka_gentestu4s_tenjob/evtgen_upsilo  
n4s_0.root | guid: 576DD4FE-C58F-8F84-6469-E93B0A322EB6 | date: 2017-06-23 05:40:24 | jobId: 5  
2280607 | site: ARC.KIT.de | status: good | size: 178872 | checksum: 3dc0c29f | checksumType: Ad  
ler32 | nEvents: 100 | experimentLow: 1 | experimentHigh: 1 | runLow: 1 | runHigh: 1 | eventLow:  
1 | eventHigh: 100 | parentGuids:  
  
52280607      Running      Job Initialization      Unknown      ARC.KIT.de
```

イベント数は100

読むほうのファイルがいっぱいある時 (解析やskim)

- ・ 今野さんのスクリプトを使ってみましょう。

```
[kiyohaya@cw05 example]$ cp ~tkonno/basf2_samples/sample04b.py .
```

- ・ 今野さんのスクリプトで使ったファイルのLFNは

```
/belle/MC/release-00-07-02/DBxxxxxxxx/MC7/  
prod00000223/s00/e0000/4S/r00000/signal/sub00/mdst  
_000001_prod00000223_task00000001.root
```

```
[kiyohaya@cw05 example]$ gb2_ds_list /belle/MC/release-00-07-02/DBxxxxxxxx/MC7/prod00000223  
/s00/e0000/4S/r00000/signal/sub00/mdst_000001_prod00000223_task00000001.root  
/belle/MC/release-00-07-02/DBxxxxxxxx/MC7/prod00000223/s00/e0000/4S/r00000/signal/sub00/mds  
t_000001_prod00000223_task00000001.root
```

じゃあ、`gbasf2 sample04b.py -p hayasaka_konno_04b -s release-00-08-00 -i /belle/MC/release-00-07-02/DBxxxxxxxx/MC7/prod00000223/s00/e0000/4S/r00000/signal/sub00/mdst_000001_prod00000223_task00000001.root` としたいところですが...

-iの挙動の違い

- basf2ではinputMdstで読み込むファイルを-iで指定できましたが、gbasf2 -i では、ファイルのディレクトリを指定します。(ファイル単体は不可)

```
[kiyohaya@cw05 example]$ gbasf2 sample04b.py -s release-00-08-00 -p hayasaka_konno_06b_sig
-i /belle/MC/release-00-07-02/DBxxxxxxxx/MC7/prod00000223/s00/e0000/4S/r00000/signal/sub00
***** Project summary *****
** Project name: hayasaka_konno_06b_sig
** Dataset path: /belle/user/kiyohaya/hayasaka_konno_06b_sig
** Steering file: sample04b.py
** Job owner: kiyohaya @ belle (23:33:39)
** Preferred site / SE: None / None
** Input files: ['/belle/MC/release-00-07-02/DBxxxxxxxx/MC7/prod00000223/s00/e0000/4S/r00000/signal/sub00/mdst_000001_prod00000223_task00000001.root']
** Number of data sets: 1
** Number of input files: 112
** Number of jobs: 112
** Processed events: 10000000 events
** Estimated CPU time per job: 30000 min
*****
Are you sure to submit the project?
Please enter Y or N: █
```

投げないでね！！

一般に1つのディレクトリに1000個
ファイルが入っています。これは信号
MCなので、112個でした。
1つのファイルを1つのジョブで
処理します。

とは言え

- いきなり 1 0 0 0 本ジョブを投げるのは危険過ぎます！
- なので、まずは 1 本のファイルだけを読むようにします。
 - まず読むファイルのメタデータを調べます。

```
[kiyohaya@cw05 example]$ gb2_ds_query_file /belle/MC/release-00-07-02/DBxxxxxxx/MC7/prod0000223/s00/e0000/4S/r00000/signal/sub00/mdst_000001_prod00000223_task00000001.root
mdst_000001_prod00000223_task00000001.root | lfn: /belle/MC/release-00-07-02/DBxxxxxxx/MC7/prod00000223/s00/e0000/4S/r00000/signal/sub00/mdst_000001_prod00000223_task00000001.root | guid: 70AF6752-DB86-1D26-8671-E9F1A1F150F2 | date: 2016-11-01 17:29:12 | jobId: 40854708 | site: LCG.KIT.de | status: good | nEvents: 90000 | experimentLow: 0 | experimentHigh: 0 | runLow: 0 | runHigh: 0 | eventLow: 500001 | eventHigh: 6140000 | parentGuids:
```

gbasf2ではメタデータを読み込むファイルの“選別条件”に使うことができます。

(将来的には、実験番号XXのRUN番号YYからZZまでという使い方をするというのを想定しています。)

GUIDを使ったファイルの選別

```
[kiyohaya@cw05 example]$ gbasf2 sample04b.py -s release-00-08-00 -p hayasaka_konno_06b_sig  
-i /belle/MC/release-00-07-02/DBxxxxxxxx/MC7/prod00000223/s00/e0000/4S/r00000/signal/sub00  
--query_file 'guid="70AF6752-DB86-1D26-8671-E9F1A1F150F2"'
```

```
***** Project summary *****  
** Project name: hayasaka_konno_06b_sig  
** Dataset path: /belle/user/kiyohaya/hayasaka_konno_06b_sig  
** Steering file: sample04b.py  
** Job owner: kiyohaya @ belle (23:26:31)  
** Preferred site / SE: None / None  
** Input files: ['/belle/MC/release-00-07-02/DBxxxxxxxx/MC7/prod00000223/s00/e0000/4S/r0000  
0/signal/sub00/mdst_000001_prod00000223_task00000001.root']  
** Processed events: 90000 events  
** Estimated CPU time per job: 30000 min  
*****
```

まだ投げないでね！！

```
Are you sure to submit the project?  
Please enter Y or N: █
```

さっきと違って112本
投げるぞいとか言って
きてない。

basf2のオプション

- basf2で-nはそこで指定した数だけイベントを読む、というものでした。
- --basf2opt はgbasf2がbasf2にオプションを渡します。
- --basf2opt="-n 100"としてやれば100イベントだけ読んで終わります。（今野さんの例題に合わせるため）

最初は、1ファイル、数イベントの読み込みでグリッド上で動くのか確認した方がいいでしょう。

今度は投げてOK

```
[kiyohaya@cw05 example]$ gbasf2 sample04b.py -s release-00-08-00 -p hayasaka_konno_06b_sig
--basf2opt=' -n 100' -i /belle/MC/release-00-07-02/DBxxxxxxx/MC7/prod00000223/s00/e0000/4S/
r00000/signal/sub00 --query file 'guid="70AF6752-DB86-1D26-8671-E9F1A1F150F2"'
*****
***** Project summary *****
** Project name: hayasaka_konno_06b_sig
** Dataset path: /belle/user/kiyohaya/hayasaka_konno_06b_sig
** Steering file: sample04b.py
** Job owner: kiyohaya @ belle (23:41:36)
** Preferred site / SE: None / None
** Input files: ['/belle/MC/release-00-07-02/DBxxxxxxx/MC7/prod00000223/s00/e0000/4S/r0000
0/signal/sub00/mdst_000001_prod00000223_task00000001.root']
** Processed events: 90000 events
** Estimated CPU time per job: 30000 min
*****
Are you sure to submit the project?
Please enter Y or N: 
```

ちゃんとプロジェクト名記録してね

さて、ジョブの状況を確認できますか

- ・ 多分、少し時間がかかるとおもいますので…

comp-users-forumに入ろう



comp-users-forum - Belle II distributed computing users' forum

comp-users-forum@belle2.org

件名: Belle II distributed computing users' forum

説明: Users' forum to discuss issues for users of the Belle II distributed computing and to make announcements to them

📧 リストのオプション

オーナー: Takanori Hara、Ueda I

モデレータ: (オーナーと同じ)

オーナーに連絡

リスト ホーム

読者登録

登録解除

投稿保管庫

投稿

共有文書置き場

<https://lists.belle2.org/sympa/info/comp-users-forum>

アカウントはconfluenceと共通

gbasf2を使っているときの疑問はここへ：英語で

comp-users-forum@belle2.org

件名: Belle II distributed computing users' forum

あなたは comp-users-forum メーリングリストへの読者登録を申し込みました。

あなたが実在の人物であることを確認します。これによって、誰かがあなたの意思に反してこのメーリングリストへの登録を申し込んでしまうのを防ぎます。これから確認用の URL を書いたメッセージをあなたが入力したメール アドレスに送ります。

メールボックスの到着メッセージを確認してください。それに書いてあるリンクをクリックしてください。これで、comp-users-forum メーリングリストへの読者登録申込を確認できます。

できあがりは？

```
[kiyohaya@cw05 example]$ gb2_ds_list hayasaka_konno_06b_sig  
/belle/user/kiyohaya/hayasaka_konno_06b_sig/sample04b_1.root
```

もちろんログも確認してね。

```
[kiyohaya@cw05 example]$ gb2_job_output -p hayasaka_konno_06b_sig  
1 project are selected.  
Please wait...
```

```
downloaded project: hayasaka_konno_06b_sig
```

```
=====  
Downloaded: "Job output sandbox retrieved in /gpfs/home/belle/kiyohaya/tutorial20170624/example/hayasaka_konno_06b_sig/52311230"
```

将来 log/プロジェクト名/ジョブ番号
になるって噂です。

ダウンロードはgb2_ds_get

```
[kiyohaya@cw05 example]$ gb2_ds_get hayasaka_konno_06b_sig
/gpfs/home/belle/kiyohaya/tutorial20170624/example/hayasaka_konno_06b_sig already exists
How would you like to verify the files? By file size(s) or by file checksum(c)?:
Please type [s] or [c]: s ← どちらでもいいです
Files to download to /gpfs/home/belle/kiyohaya/tutorial20170624/example/hayasaka_konno_06b_
sig :
/belle/user/kiyohaya/hayasaka_konno_06b_sig/sample04b_1.root
Do you want to download files:
Please type [Y] or [N]: y

Download 1 files from SE
__getFile: Using 1 streams
__getFile: Executing transfer of srm://gridka-dcache.fzk.de:8443/srm/managerv2?SFN=/pnfs/gr
idka.de/belle/disk-only/TMP/belle/user/kiyohaya/hayasaka_konno_06b_sig/sample04b_1.root to
file: /gpfs/home/belle/kiyohaya/tutorial20170624/example/hayasaka_konno_06b_sig/sample04b_1.
root

Successfully downloaded files:
/belle/user/kiyohaya/hayasaka_konno_06b_sig/sample04b_1.root in /gpfs/home/belle/kiyohaya/t
utorial20170624/example/hayasaka_konno_06b_sig
```

果たして…

basf2の方の窓でrootを起動。(gbasf2の環境ではroot呼べない)

```
root [0] TFile *f = new TFile("sample04b.root");
root [1] b0->Scan("evt_no");
```

```
*****
*      Row      *      evt_no *
*****
*          0 *      700001 *
*          1 *      700002 *
*          2 *      700003 *
*          3 *      700004 *
*          4 *      700014 *
*          5 *      700017 *
*          6 *      700021 *
*          7 *      700025 *
*          8 *      700034 *
*          9 *      700039 *
*         10 *      700039 *
*         11 *      700044 *
*         12 *      700050 *
*         13 *      700053 *
*         14 *      700055 *
*         15 *      700059 *
*         16 *      700061 *
*         17 *      700063 *
*         18 *      700066 *
*         19 *      700068 *
*         20 *      700074 *
*         21 *      700075 *
*         22 *      700075 *
*         23 *      700076 *
*         24 *      700080 *
```

```
root [2] TFile *g = new TFile("../example/hayasaka_konno_06b");
root [3] b0->Scan("evt_no");
```

```
*****
*      Row      *      evt_no *
*****
*          0 *      700001 *
*          1 *      700002 *
*          2 *      700003 *
*          3 *      700004 *
*          4 *      700014 *
*          5 *      700017 *
*          6 *      700021 *
*          7 *      700025 *
*          8 *      700034 *
*          9 *      700039 *
*         10 *      700039 *
*         11 *      700044 *
*         12 *      700050 *
*         13 *      700053 *
*         14 *      700055 *
*         15 *      700059 *
*         16 *      700061 *
*         17 *      700063 *
*         18 *      700066 *
*         19 *      700068 *
*         20 *      700074 *
*         21 *      700075 *
*         22 *      700075 *
*         23 *      700076 *
*         24 *      700080 *
```

query_file もうちょっと

```
[kiyohaya@cw05 example]$ gb2_ds_list /belle/MC/release-00-07-02/DBxxxxxxxx/MC7/prod00000223/s00/e0000/4S/r00000/signal/sub00/
/belle/MC/release-00-07-02/DBxxxxxxxx/MC7/prod00000223/s00/e0000/4S/r00000/signal/sub00/mdst_000001_prod00000223_task00000001. r
oot
/belle/MC/release-00-07-02/DBxxxxxxxx/MC7/prod00000223/s00/e0000/4S/r00000/signal/sub00/mdst_000002_prod00000223_task00000002. r
oot
/belle/MC/release-00-07-02/DBxxxxxxxx/MC7/prod00000223/s00/e0000/4S/r00000/signal/sub00/mdst_000003_prod00000223_task00000003. r
oot
/belle/MC/release-00-07-02/DBxxxxxxxx/MC7/prod00000223/s00/e0000/4S/r00000/signal/sub00/mdst_000004_prod00000223_task00000004. r
oot
/belle/MC/release-00-07-02/DBxxxxxxxx/MC7/prod00000223/s00/e0000/4S/r00000/signal/sub00/mdst_000005_prod00000223_task00000005. r
oot
/belle/MC/release-00-07-02/DBxxxxxxxx/MC7/prod00000223/s00/e0000/4S/r00000/signal/sub00/mdst_000006_prod00000223_task00000006. r
oot
/belle/MC/release-00-07-02/DBxxxxxxxx/MC7/prod00000223/s00/e0000/4S/r00000/signal/sub00/mdst_000007_prod00000223_task00000007. r
oot
/belle/MC/release-00-07-02/DBxxxxxxxx/MC7/prod00000223/s00/e0000/4S/r00000/signal/sub00/mdst_000009_prod00000223_task00000009. r
oot
/belle/MC/release-00-07-02/DBxxxxxxxx/MC7/prod00000223/s00/e0000/4S/r00000/signal/sub00/mdst_000010_prod00000223_task00000010. r
oot
```

- 何故か 8 が無いのはよくわかりませんが...

lfn LIKE という選別条件

```
[kiyohaya@cw05 example]$ gbasf2 sample04b.py -s release-00-08-00 -p hayasaka_konno_06b_sig2 -i /belle/MC/release-00-07-02/DBxxxxxxx/MC7/prod00000223/s00/e0000/4S/r00000/signal/sub00 --query_file "lfn LIKE '%mdst_00000%'"
*****
***** Project summary *****
** Project name: hayasaka_konno_06b_sig2
** Dataset path: /belle/user/kiyohaya/hayasaka_konno_06b_sig2
** Steering file: sample04b.py
** Job owner: kiyohaya @ belle (22:27:37)
** Preferred site / SE: None / None
** Input files: ['/belle/MC/release-00-07-02/DBxxxxxxx/MC7/prod00000223/s00/e0000/4S/r00000/signal/sub00/mdst_000001_prod00000223_task00000001.root']
** Number of data sets: 1
** Number of input files: 8
** Number of jobs: 8
** Processed events: 720000 events
** Estimated CPU time per job: 30000 min
*****
Are you sure to submit the project?
Please enter Y or N: █
```

ここでは%が
ワイルドカード

← 8本のジョブが投入される

投げないでね！！

gbasf2 の -n

- signalMCもバラバラってのもねえという人も居るでしょう

-nはgbasf2で読み込むfile数を指定できます

```
[kiyohaya@cw05 example]$ gbasf2 sample04b.py -s release-00-08-00 -p hayasaka_konno_06b_sig2 -n 8 -i /belle/MC/release-00-07-02/DBxxxxxxx/MC7/prod00000223/s00/e0000/4S/r000000/signal/sub00 --query_file "lfn LIKE '%mdst_00000%'"
```

```
*****
```

```
***** Project summary *****
```

```
** Project name: hayasaka_konno_06b_sig2
```

```
** Dataset path: /belle/user/kiyohaya/hayasaka_konno_06b_sig2
```

```
** Steering file: sample04b.py
```

```
** Job owner: kiyohaya @ belle (22:24:18)
```

```
** Preferred site / SE: None / None
```

```
** Input files: ['/belle/MC/release-00-07-02/DBxxxxxxx/MC7/prod00000223/s00/e0000/4S/r00000/signal/sub00/mdst_000001_prod00000223_task00000001.root', '/belle/MC/release-00-07-02/DBxxxxxxx/MC7/prod00000223/s00/e0000/4S/r00000/signal/sub00/mdst_000002_prod00000223_task00000002.root', '/belle/MC/release-00-07-02/DBxxxxxxx/MC7/prod00000223/s00/e0000/4S/r00000/signal/sub00/mdst_000003_prod00000223_task00000003.root', '/belle/MC/release-00-07-02/DBxxxxxxx/MC7/prod00000223/s00/e0000/4S/r00000/signal/sub00/mdst_000004_prod00000223_task00000004.root', '/belle/MC/release-00-07-02/DBxxxxxxx/MC7/prod00000223/s00/e0000/4S/r00000/signal/sub00/mdst_000005_prod00000223_task00000005.root', '/belle/MC/release-00-07-02/DBxxxxxxx/MC7/prod00000223/s00/e0000/4S/r00000/signal/sub00/mdst_000006_prod00000223_task00000006.root', '/belle/MC/release-00-07-02/DBxxxxxxx/MC7/prod00000223/s00/e0000/4S/r00000/signal/sub00/mdst_000007_prod00000223_task00000007.root', '/belle/MC/release-00-07-02/DBxxxxxxx/MC7/prod00000223/s00/e0000/4S/r00000/signal/sub00/mdst_000009_prod00000223_task00000009.root']
```

```
** Processed events: 720000 events
```

```
** Estimated CPU time per job: 240000 min
```

← Input Filesが8つ

投げないでね！！

-n 4を指定すれば

```
[kiyohaya@cw05 example]$ gbasf2 sample04b.py -s release-00-08-00 -p hayasaka_konno_06b_sig2 -n 4 -i /belle/MC/release-00-07-02/DBxxxxxxx/MC7/prod00000223/s00/e0000/4S/r00000/signal/sub00 --query_file "lfn LIKE '%mdst_00000%'"
*****
***** Project summary *****
** Project name: hayasaka_konno_06b_sig2
** Dataset path: /belle/user/kiyohaya/hayasaka_konno_06b_sig2
** Steering file: sample04b.py
** Job owner: kiyohaya @ belle (22:21:47)
** Preferred site / SE: None / None
** Input files: ['/belle/MC/release-00-07-02/DBxxxxxxx/MC7/prod00000223/s00/e0000/4S/r00000/signal/sub00/mdst_000001_prod00000223_task00000001.root', '/belle/MC/release-00-07-02/DBxxxxxxx/MC7/prod00000223/s00/e0000/4S/r00000/signal/sub00/mdst_000002_prod00000223_task00000002.root', '/belle/MC/release-00-07-02/DBxxxxxxx/MC7/prod00000223/s00/e0000/4S/r00000/signal/sub00/mdst_000003_prod00000223_task00000003.root', '/belle/MC/release-00-07-02/DBxxxxxxx/MC7/prod00000223/s00/e0000/4S/r00000/signal/sub00/mdst_000004_prod00000223_task00000004.root']
** Number of data sets: 1
** Number of input files: 8
** Number of jobs: 2
** Processed events: 720000 events
** Estimated CPU time per job: 120000 min
*****
```

Input Filesが4つ

Jobの数が2つ

投げないでね！！

最後に

- グリッドジョブは慎重に準備しないと期待通りに動いてくれないことが多いので、面倒でも慎重に準備してください。
- 例えば、ログのサイズは～10Mbytesなので、十分に確認が取れたらログの出力は最小限になるように設定するようにしてください。
 - イベントのダンプなんかも、最小限のデータセットを取り出して、そいつをダウンロードして、ローカルでダンプするなど工夫を。
- 昨日、T原さんが、gbasf2講習会をやりました。（英語）

<https://kds.kek.jp/indico/event/24563/session/45/?slotId=0#20170623>

hop, step, jumpというスライドがあります。

（jumpは自分で作ったmoduleをgridで実行する話）

今日の話聞いた後だと、よりわかりやすく読めると思います。

BACKUP

グリッド証明書ファイルの取得

- KEKの手続きが済んでKEKからお手紙が来たら…
- KEKCCにログインします。

```
/opt/kek/cactl/bin/certreq issue -uid <KEKからのお手紙  
に書いてあるユーザー名> -ucert
```

ここで、**Input Challenge PIN or Password :**

とパスワードが聞かれますがこれは、お手紙に書いてあるパスワードかそれに対応する変更したパスワードです。

この後、

Input PASS Phrase:

と聞かれますが、これは上記とは別のもので、今後グリッドを使う度に入力するものですので、長いものを新たに決めてください。

証明書ファイル

- usercert.pem, userkey.pem, 617ff41b.0

の3つのファイルができていますので、なくさないようにしてください。保存は他人が入れないよう `chmod 700` したディレクトリにおいておいてください。

これを変換して、browserで読み込める形にします。

```
openssl pkcs12 -export -in usercert.pem -inkey userkey.pem -out MyCert.p12
```

この時、**Enter pass phrase for userkey.pem** と聞かれますが、これは先ほど新たに決めたpass phraseです。

その後、**Enter Export Password**と言われますが、これはまた改めて決めてください。これは、このp12ファイルをブラウザにインストールする時に聞かれます。

MyCertの部分は好きな名前にしてください。毎年更新するので、区別するためにファイル名に年を入れておくのをお勧めします。

このファイルも他人には読めないようにしてください。

```
chmod 400 MyCert.p12
```

証明書ファイルのブラウザへのインストール

- KEKCCから各自のノートパソコンへのファイルのコピーは自分で…コピーするのはp12だけです。
- マックはp12ファイルをダブルクリックでインストールできる？
(未確認)
- Firefoxは右上の≡ (三本線) →オプション→詳細→証明書→証明書を表示→あなたの証明書→インポート
- Chromeは右上の点3つ→設定→詳細設定を表示 (一番下) →HTTPS/SSL 証明書の管理→個人→インポート

Belle VO に登録

- <https://voms.cc.kek.jp:8443/voms/belle>

に行く。

- Registration をクリックして、項目を埋めて登録
(メールアドレスを絶対間違えないように)
- 上で登録したアドレスにメールが届くので、メールの本文に記載されているURLに飛ぶ
- また、フォームが出てくるので、情報を埋める
- その後、登録完了のメールが来る (ここは人手なので時間がかかるかも)

DIRACへの登録

- 下記のフォーマットでcomp-dirac-admin@belle2.orgにメールを送る

Name : John Doe

KEKCC account : johndoe

**DN : /DC=XXX/DC=yyyy/O=ZZZ/OU=aaa/CN=John Doe
bbb**

email : john.doe@your.institute

※自分のDNは

<https://voms.cc.kek.jp:8443/voms/belle/info/index.action>

にアクセスするとsubjectとして出てくる文字列です。

メールの件名は決まってませんが「DIRAC registration」あたりでお願いします。