

# US Belle II Summer School: Preparations for basf2 tutorials

Jake Bennett  
Carnegie Mellon University  
August, 2016

# Prerequisites - Accounts

- In order to fully participate in the tutorials, you must ensure that you have all of the necessary accounts and permissions
- Follow (or verify) all the steps in the [Belle II registration procedure](#)
  - You must obtain a grid certificate,
  - load it into your browser,
  - join the belle VO (virtual organization),
  - [register with DIRAC](#),  
(you really only need steps 5 and 6 on this page, steps 1-4 are redundant with the steps on the Belle II registration page)
  - and apply for a DESY account.  
(Note: you need not request access to stash, this is now done by default.)
- **BE AWARE** that this process can take several days!  
Please initiate this procedure as soon as possible!

# Prerequisites - Resources

- The KEKCC schedule shows a brief shutdown in early August
  - The storage area holding MC5 and MC6 samples will be taken offline and will be unavailable during the summer school
  - *It should be possible to use KEKCC resources during the summer school*
- **However, it is useful to have basf2 prepared on your own machine, so we will use this as a backup**
  - Still useful, I do most of my development on a local repository anyway
  - Analysis directories are transferable (can be accessed anywhere)

# Prerequisites - Resources

- The following slides are meant to guide you through setting up basf2 on your own computing resources
- CAUTION: Only Scientific Linux 6 and Ubuntu 14.04 are officially supported for basf2. Other systems may be used, but binary tarballs are only available for the officially supported systems.
- **My recommendation:** If you do not have easy access to computing resources (via ssh for example), use a laptop and a virtual desktop (like [Oracle VirtualBox](#)) for the tutorials
  - *Be sure to prepare early, it can take time to download and install*
  - You can find step by step instructions [below](#) (do this first)
- You can also [set up an ssh key pair](#) for ease of access to DESY's stash

## Some useful information

- I have detailed the installation procedure in the following slides, but there is still a good bit of development ongoing. You can check the confluence pages for up to date information. You can also contact the git support team at [git-support@belle2.org](mailto:git-support@belle2.org)
- The following pages review the procedures detailed in the following and go into a little more detail
  - [Registration procedure](#)
  - [Software installation](#)
  - [Git/Stash introduction](#)
  - [Git introduction](#)
- You can also navigate from those pages to find additional details
- **Please contact me if you have questions or comments ([jbennett@cmu.edu](mailto:jbennett@cmu.edu))**

# Prerequisites - Getting started

- Before we get started, we need to **download and install the basic development packages** (like git) that are needed to build the software
- Open a terminal and create and enter a basf2 directory, then download the prepare\_belle2.sh script from the git repository

```
[~build-2016-07-05]$ mkdir basf2; cd basf2  
[~basf2]$ curl --user <username> -G "https://stash.desy.de/projects/B2/  
repos/tools/browse/prepare_belle2.sh?raw" > prepare_belle2.sh
```

- Next, make prepare\_belle2.sh executable and run it

```
[~basf2]$ chmod a+x prepare_belle2.sh  
[~basf2]$ ./prepare_belle2.sh
```

- Now, you should be ready to set up basf2!

## Prerequisites - Accessing stash

- Verify that you can access the code repository  
(here and elsewhere: replace “username” with your DESY account name)

```
[~basf2]$ git ls-remote https://username@stash.desy.de/scm/b2/tools.git master
```

- If you have uploaded your ssh key to stash, you can instead use

```
[~basf2]$ git ls-remote ssh://git@stash.desy.de:7999/b2/tools.git master
```

- If you get a message like the following

```
The authenticity of host '[stash.desy.de]:7999 ([131.169.180.180]:7999)' can't be established.  
RSA key fingerprint is 0d:ce:90:19:3f:94:bd:d6:50:2e:fb:ee:cc:50:0c:6b.  
Are you sure you want to continue connecting (yes/no)?
```

- Select yes after verifying the fingerprint
- If you successfully connected to the repository, you should see something like

```
5a8c4c0ef787fc6e47e9c96bfd73e526c6c3e78a refs/heads/master
```

## Prerequisites - Accessing stash

- By default, the software tools will use your username to access stash
- If your DESY account name is different from your local user name, set the BELLE2\_USER environment variable

```
[~basf2]$ export BELLE2_USER=username
```

- TIP: You can make your own git repository on stash by following the procedures in the [git/stash introduction](#)
  - Not necessary for the summer school, but it is a useful feature!
  - Share version controlled documents or scripts with colleagues

# Prerequisites - Installing the software

- *If you are working on a machine that has a central release, you can use it instead of doing a full installation, which is what the rest of us will be using*

- First, we need to **install the software tools** (creates a directory “tools”)

```
[~basf2]$ git clone ssh://git@stash.desy.de:7999/b2/tools.git
```

- If you have not installed the ssh key pair, use

```
[~basf2]$ git clone https://username@stash.desy.de/scm/b2/tools.git
```

- Now, set up your environment for Belle II

```
[~basf2]$ source tools/setup_belle2
```

- If the version of tools is out of date, move into the tools directory and update

```
[~basf2]$ cd tools; git pull --rebase; cd ..
```

# Prerequisites - Installing the software

- Next, you need to **install some third-party code** (called externals), like ROOT and GEANT4, used by the Belle II software
- You can check the available versions of the externals

```
[~basf2]$ get_externals.sh
```

- We will use version v01-01-08 of the externals

```
[~basf2]$ get_externals.sh v01-01-08
```

- If you are working on a supported OS, you can get a precompiled version

```
[~basf2]$ get_externals.sh v01-01-08 ubuntu1404
```

- This can take quite a long time.  
You should take a coffee break.



# Prerequisites - Installing the software

- When the download is finished (as well as your coffee), you are finished installing supporting software. Now we will install basf2.
- Since we will be working with a specific release of the software (release-00-07-01), we need to install the release “centrally”.  
If you are using a supported OS, you can get a precompiled version

```
[~basf2]$ get_release.sh release-00-07-01 ubuntu1404
```

Otherwise,

```
[~basf2]$ get_release.sh release-00-07-01
```

- Another coffee break seems appropriate



# Prerequisites - Validating the software

- Now set up the basf2 environment (from the central release)

```
[~basf2]$ setuprel release-00-07-01
```

- Do a quick test to see if everything was properly installed

```
[~basf2]$ basf2 --info
```

- You can also list the available modules

```
[~basf2]$ basf2 -m
```

Or details about a specific module

```
[~basf2]$ basf2 -m RootInput
```

- If you do not see any errors, you are ready to begin using basf2!

Helpful, but not necessary, details

# Setting up a local release

- We will not use this during the school, but it is useful nonetheless
- If you want to work on basf2 code development, you will need to create a local release

```
[~basf2]$ newrel release
```

This creates the 'release' folder, checks out all default packages, and sets up the build scripts for the head version of the code

- If you want to use the head version of some packages while using others from a specific release, you can specify the central release name

```
[~basf2]$ newrel release release-00-07-01
```

(Note, the first argument is the directory name, which need not be 'release')

## Setting up a local release

- No packages are checked out, but you can add packages with the ‘*addpkg package*’ command

```
[~basf2]$ cd release  
[~release]$ setuprel  
[~release]$ addpkg analysis`
```

- Now compile the code

```
[~release]$ scon
```

# Creating an SSH key pair

- First, create the private and public keys and create a password

```
[~basf2]$ ssh-keygen -t rsa -C "your_email@your_domain.com"
```

(Note: this will create two files, **id\_rsa is the private part - DO NOT SHARE!** - id\_rsa.pub is the public part - you can share and upload it)

- Upload the key pair to Stash by opening a browser to <https://stash.desy.de/plugins/servlet/ssh/account/keys> and logging in with your DESY account
- Select “Add Key” and copy the contents of id\_rsa.pub into the box, then click “Add Key”
- Check your access

```
[~basf2]$ git ls-remote ssh://git@stash.desy.de:7999/b2/tools.git master
```

- You should see something like this

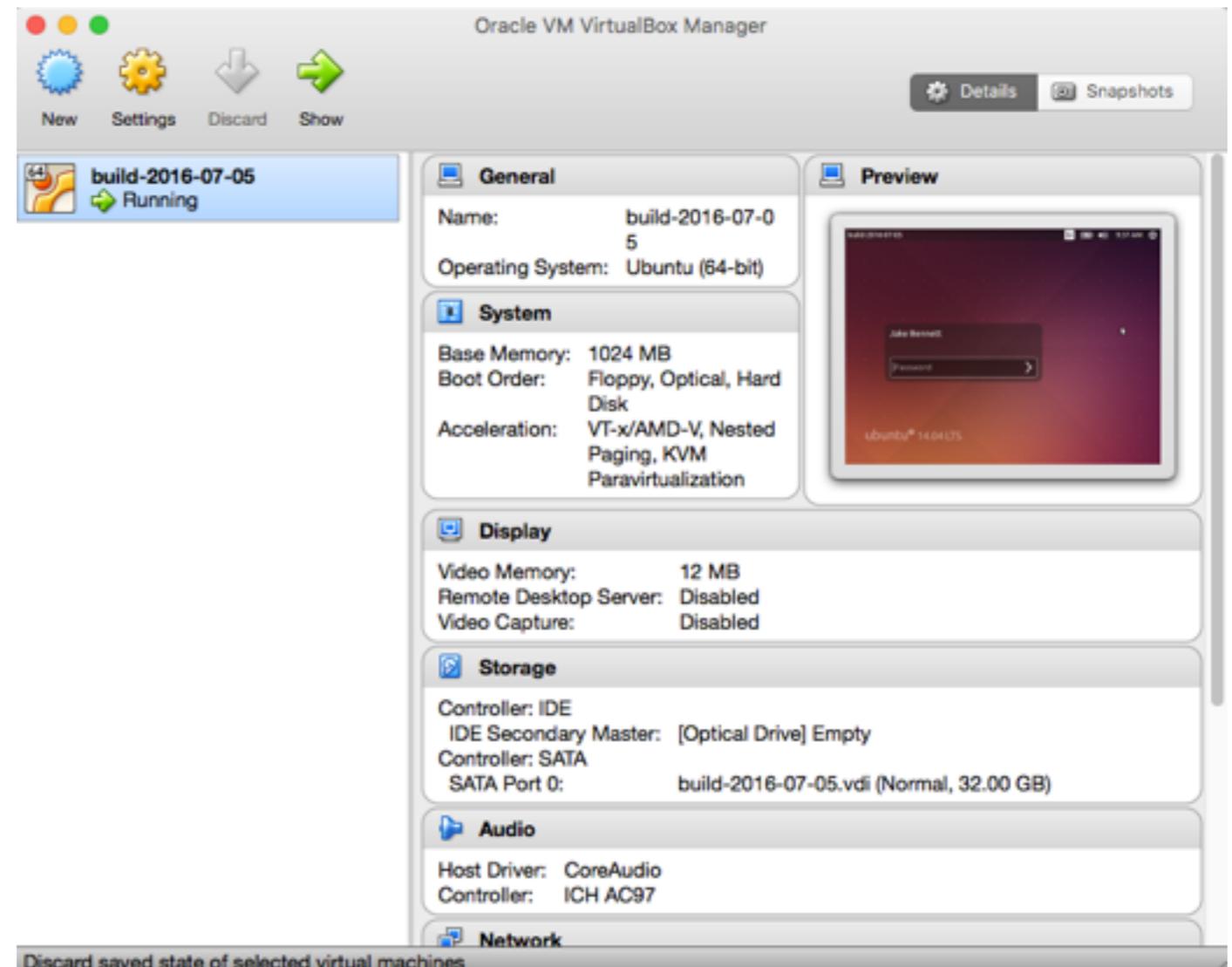
```
5a8c4c0ef787fc6e47e9c96bfd73e526c6c3e78a refs/heads/master
```

# Setting up VirtualBox

- [Download the program](#) and install it on your machine
- Download the [Ubuntu 14.04 desktop image](#)
- Run VirtualBox and click “New” (top left corner)
- Name your virtual machine (I named mine “build-2016-07-05” since we will be using the basf2 build with that name), select “Linux” under type, the version should be “Ubuntu (64-bit)”: hit “Continue”
- Depending on the memory of your machine, you can just use the default memory allocation (I bumped mine up to 1 GB)
- Create a virtual hard disk (default option) by clicking “Create”
  - Leave the default “VDI” selected and hit “Continue”
  - Similarly for “Dynamically allocated”
  - Increase the size of your virtual hard disk to 30 GB and hit “Create”
- Now the hard disk you just created should show up on the panel on the right

# Setting up VirtualBox

- Double click the icon of your virtual hard drive
- Click the folder icon beside the drop down window and find the iso file for your download of Ubuntu 14.04 (ubuntu-14.04.4-desktop-amd64.iso, probably in your “Downloads” directory)
- Follow the on screen directions to install Ubuntu onto your virtual hard drive
- After installation is complete, follow the directions to get started with basf2 (Hint: use the search function in the top left application to find the “terminal” application)



# Setting up VirtualBox

- Here are some settings I have found useful while working with VirtualBox
  - “Devices” -> “Shared Clipboard” -> “Bidirectional” (allows for copy and paste from your OS to the VirtualBox OS)
  - “Devices” -> “Insert Guest Additions CD image...” (allows you to scale the window of your virtual OS - requires restart)