Document Key:

(These are variables used throughout this document to describe locations and studio specific codes)

SOFTWARE_ROOT = “\\your\install\software\packages”
(Root location of studios software)

RIG_BUILDER_PROJECT_ROOT = “Some project location”
(This is the location where all rigging related files will be saved. This should be outside and separate to any shotgun related directories)

VERSION_ROOT = “A software packages latest version directory”.
(This is the version directory of a software package. eg(0.0.16 might be the latest version of rigBuilder package, and look something like “$SOFTWARE_ROOT/rigBuilder/0.0.16”))

note: When the symbol $ followed by () is used it represents a variable for one of the above keys.
eg. $(SOFTWARE_ROOT) ← This is the variable for the value ‘SOFTWARE_ROOT’

RigBuilder Project Creation

(This will guide you through setting up a project for rigBuilder so that it knows where to load and publish files to)

The RigBuilder needs a location to save and publish files to. These files are not related to Honi or Shotgun in any way. The directory in which your project lives is completely up to you. However it should not be within Shotgun’s publishing locations as it is separate to Shotgun’s directory structure.

In this example I will use an example directory ‘S:\tim.mackintosh\rigBuilderProject’ as my RIG_BUILDER_PROJECT_ROOT. So this needs to be set as the value of the key ‘RIG_BUILDER_PROJECT_ROOT’ in the json file that lives here, “$(SOFTWARE_ROOT)/rigBuilder/($VERSION_ROOT)/dingo.json”.

For more information on setting this value refer to the document ‘RigBuilder Sofware Package Setup’.

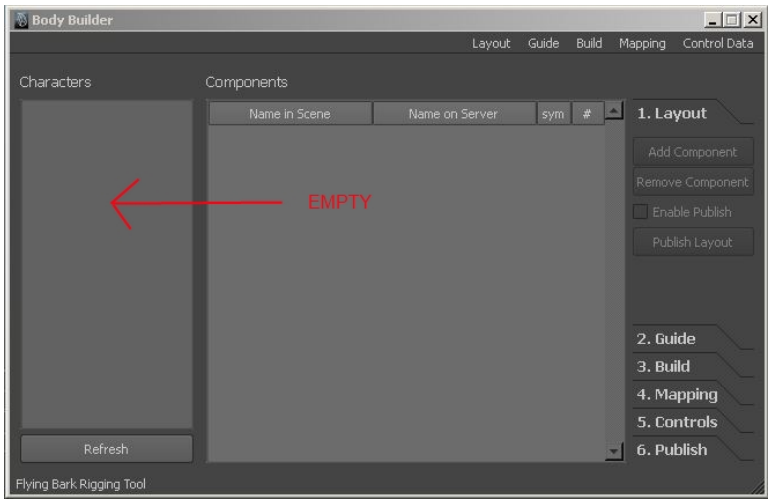
Creating the base asset and character directories for your project:

In your “$(RIG_BUILDER_PROJECT_ROOT)” directory create another directory called assets. Under the new ‘asset’ directory create another directory called character.

The file structure should now look like this:

$(RIG_BUILDER_PROJECT_ROOT)
|
asset
|
character

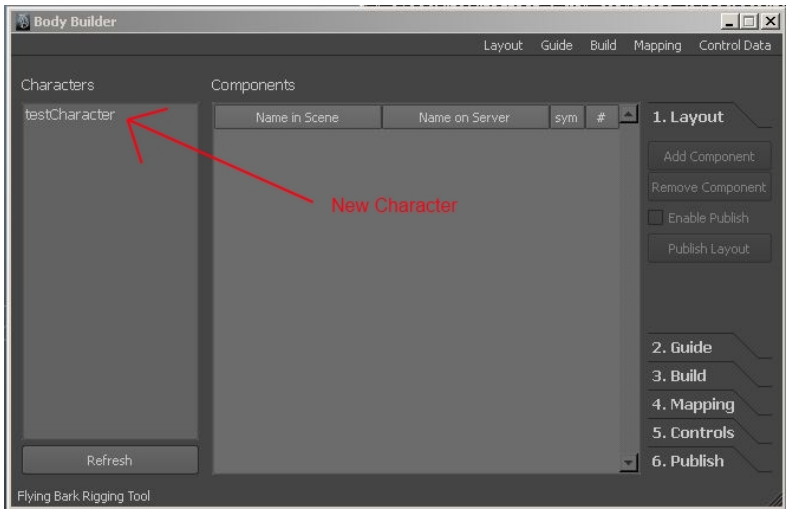
The ‘character’ directory will be the base directory for your individual characters. By default you will not have any characters and the ‘Body Builder’ ui will be empty and look like this.



In order to to create a character for the rigBuilder to find we need to create another directory under the ‘character’ directory that we created previously. So to create the character called ‘testCharacter’ we need to create a directory under ‘character’ called ‘testCharacter’. After doing this your directory structure should look like this:

$(RIG_BUILDER_PROJECT_ROOT)
|
asset
|
character
|
testCharacter

Now if you hit the refresh button on the lower left of the ‘Body Builder’ window you will see ‘testCharacter’ under the ‘characters’ list on the left hand side, like this:



Copying the base components for the 'Body Builder' from package to project:

(A default set of components comes with the rigBuilder package and needs to be copied into your project)

In order to create creatures, components need to be developed such as arm, leg and torso components. Flying Bark will deliver a default set of components for the base of bipeds, quadrupeds and birds in the rigBuilder package. These default components need to be copied to your "$(RIG_BUILDER_PROJECT_ROOT)/asset" directory.

From your "$(VERSION_ROOT)/resource/body" directory. Copy the 'component' directory to your "$(RIG_BUILDER_PROJECT_ROOT)/asset " directory.

Your directory structure should now look like this using 'S:\tim.mackintosh\rigBuilderProject' as an example for the RIG_BUILDER_PROJECT_ROOT variable: