3d Brain Atlas Reconstructor Installation (Ubuntu)

Note: This procedure is valid for *Ubuntu 9.04*, *Ubuntu 10.04 LTS*, *Ubuntu 10.10* and *Ubuntu 11.04*. Installation on other Ubuntu versions or other Linux distributions is similar but the packages versions may be slightly different.

Getting the code

It is assumed that the main directory dedicated for 3dBAR is `/home/$USERNAME/3dbar`. if you want to install it in another directory, replace `3dbar` with the desired path.

To get the latest stable version of 3dBAR fill out the following form then download 3dBAR using the link provided via email.

Unzip the file to your home directory and go to the 3dBAR directory:

```
mkdir ~/3dbar; unzip 3dbar.zip -d ~/3dbar ; cd ~/3dbar;
```

The directories have the following purposes:

- **bin**: Holds all executable files, atlas parsers and auxiliary scripts
- **lib**: Holds the 3dBAR api
- **atlases**: Directory, where the source data, *CAF datasets* and reconstructed models are stored. Each dataset (denoted as DATASET_NAME) contains the following subdirectories:
    - atlases/DATASET_NAME/src : Here the source data is located. It can be placed manually by a user or downloaded from internet depending on a particular parser.
    - atlases/DATASET_NAME/caf : This is the directory where a CAF dataset is generated by specific parsers.
    - atlases/DATASET_NAME/reconstructions : The directory for reconstructed models.

Then follow instructions from `README` file to verify if the installation was successful.

Installing required packages

Installation in Ubuntu 10.10 and Ubuntu 11.04

1. Install the Visualization Toolkit and other graphics libraries:

```
sudo apt-get install \
libvtk5.4 libvtk5-dev libvtk5.4-qt4 libvtk5-qt4-dev \
tk8.5 tk8.5-dev \
python-vtk libgtkgl2.0-1 libgtkgl2.0-dev libgtkglext1 librsvg2-2 python-nifti
```

2. Install python related packages:

```
sudo apt-get install \
python-gtkglext1 python-rsvg python-opengl python-numpy python-scipy python-wxgtk2.8
```

3. Other packages:

```
sudo apt-get install \
potrace pstoedit python-setuptools python-epydoc
```

Installation in Ubuntu 12.04 LTS

1. Install the Visualization Toolkit and other graphics libraries:

```
sudo apt-get install \
libvtk5.8 libvtk5-dev libvtk5.8-qt4 libvtk5-qt4-dev \
tk8.5 tk8.5-dev \
python-vtk libgtkgl2.0-1 libgtkgl2.0-dev libgtkglext1 librsvg2-2 python-nifti
```

2. Install python related packages:

```
sudo apt-get install \
python-gtkglext1 python-rsvg python-opengl python-numpy python-scipy python-wxgtk2.8
```

3. Other packages:

```
sudo apt-get install \
potrace pstoedit python-setuptools python-epydoc
```

Installation in Ubuntu 11.10

1. Install the Visualization Toolkit and other graphics libraries:

```
sudo apt-get install \
libvtk5.6 libvtk5-dev libvtk5.6-qt4 libvtk5-qt4-dev \
tk8.5 tk8.5-dev \
python-vtk libgtkgl2.0-1 libgtkgl2.0-dev libgtkglext1 librsvg2-2 python-nifti
```

2. Install python related packages:

```
sudo apt-get install \
python-gtkglext1 python-rsvg python-opengl python-numpy python-scipy python-wxgtk2.8
```

3. Other packages:

```
sudo apt-get install \
potrace pstoedit python-setuptools python-epydoc
```

Installation in Ubuntu 10.04

1. Install the Visualization Toolkit and other graphics libraries:

```
sudo apt-get install \
libvtk5.2 libvtk5-dev libvtk5.2-qt4 libvtk5-qt4-dev \
tk8.5 tk8.5-dev \
python-vtk libgtkgl2.0-1 libgtkgl2.0-dev libgtkglext1 librsvg2-2 python-nifti
```

2. Install python related packages:

```
sudo apt-get install \
python-gtkglext1 python-rsvg python-opengl python-numpy python-scipy python-wxgtk2.6
```

3. Other packages:

```
sudo apt-get install \
potrace pstoedit python-setuptools python-epydoc
```

Installation in Ubuntu 9.10

1. Install the Visualization Toolkit and other graphics libraries:

```
sudo apt-get install \
libvtk5.2 libvtk5-dev libvtk5.2-qt4 libvtk5-qt4-dev \
tk8.5 tk8.5-dev \
python-vtk libgtkgl2.0-1 libgtkgl2.0-dev libgtkglext1 librsvg2-2 python-nifti
```

2. Install python related packages:

```
sudo apt-get install \
python-gtkglext1 python-rsvg python-opengl python-numpy python-scipy python-wxgtk2.6
```

3. Other packages:

```
sudo apt-get install \
potrace pstoedit python-setuptools python-epydoc
```

If you are a developer you may also want to install optional packages with documentation:

```
sudo apt-get install vtkdata vtk-doc vtk-examples
```

Installation in Ubuntu 8.04

Installation consists of following steps (just paste code blocks into terminal it should be fine:

1. Installing Visualization Toolkit and other graphic libraries:

```
sudo apt-get install \
libvtk5 libvtk5-dev libvtk5-qt4 libvtk5-qt4-dev \
tk8.5 tk8.5-dev \
python-vtk libgtkgl2.0-1 libgtkgl2.0-dev libgtkglext1 librsvg2-2 python-nifti
```

2. Installing python-related packages:

```
sudo apt-get install \
python-gtkglext1 python-opengl python-numpy python-scipy \
python-gnome2 python-gnome2-desktop python-gnome2-desktop-dev python-gnome2-dev python-wxgtk2.6
```

3. Other packages:

```
sudo apt-get install \
potrace pstoedit python-setuptools subversion python-epydoc
```

If You are a developer, you may also want to install optional packages with documentation:

```
sudo apt-get install vtkdata vtk-doc vtk-examples
```

Troubleshooting

Segmentation fault in Ubuntu 11.10

If the reconstructor crashes like that (numbers can vary):

```
$ ./3dbar.sh
./3dbar.sh: line 17:  2296 Segmentation fault      python bin/reconstructor/gui.py
```

the reason can be a bug in the 'python-vtk' package installed in your system. Unfortunately there is no automated way to fix it - you have to do it manually:

1. Find a file named 'wxVTKRenderWindowInteractor.py'. It can be located in '/usr/share/pyshared/vtk/wx/' directory or in similar location:

```
$ find / -name 'wxVTKRenderWindowInteractor.py'
```

2. Edit the file with your favourite ASCII editor. In the example editor 'vim' is used and it is assumed that the path to the file is '/usr/share/pyshared/vtk/wx/wxVTKRenderWindowInteractor.py':

```
$ sudo vim /usr/share/pyshared/vtk/wx/wxVTKRenderWindowInteractor.py
```

3. Near 350th line of the file find a following line:

```
d = '_%s_%s' % (d[2:], 'void_p')
```

4. Add '\0' characters to the line to make it like below:

```
d = '_%s_%s\0' % (d[2:], 'void_p')
```

5. Save the modified file.

6. The bug should be fixed for now. Try running 3dBAR again. If this solution doesn't work - let us know.