Creating an Animation from GPlates Output Using FFmpeg

Authors: Christopher Alfonso, Sabin Zahirovic
EarthByte Research Group, School of Geosciences, University of Sydney, Australia

Aim

This tutorial is designed to (1) teach users how to export screenshots of plate reconstructions as discrete image files and then turn these image files into an animated video file, and (2) assist users with the installation of the software required to do so.

Background

GPlates allows you to visualise plate reconstructions and raster data through geological time. This data can then be exported and converted to file formats, including images and videos, which can be viewed independently of

GPlates and can be useful for creating figures and presentations.

With Exercise 1, we will begin by exporting a series of images from GPlates, recapping and building upon concepts covered in the previous tutorial (Tutorial 6.2: Exporting Plate Reconstructions as Image Files). We will then move on to cover the process of converting these images into an animated video file in Exercise 2.

Relevant Files

This exercise will use the GPlates sample data included with the GPlates installation – specifically, the "Data Bundle for Novices" GPlates project file, which can be found at the following location within the GPlates installation directory:

SampleData\DataBundleForNovices\DataBundleForNovices.gproj

Exercise 1 – Exporting a series of reconstructions and including a timestamp in GPlates

---

For this exercise, we will load the GPlates "Data Bundle for Novices" project file, along with its accompanying feature collection files, using "File" → "Open Project…":

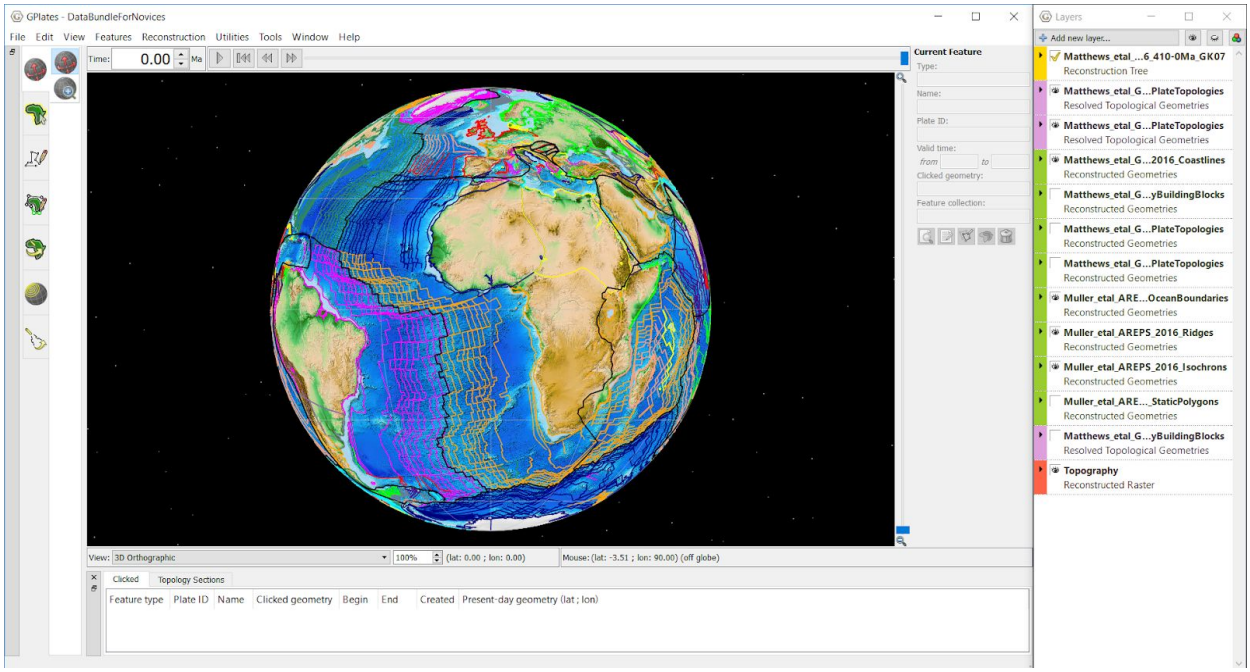

Figure 1. GPlates with the appropriate datasets loaded.

To simplify things, we can turn off all of the layers except for the resolved topologies, coastlines, and topography by clicking the eyes next to their names in the “Layers” window (Figure 2).

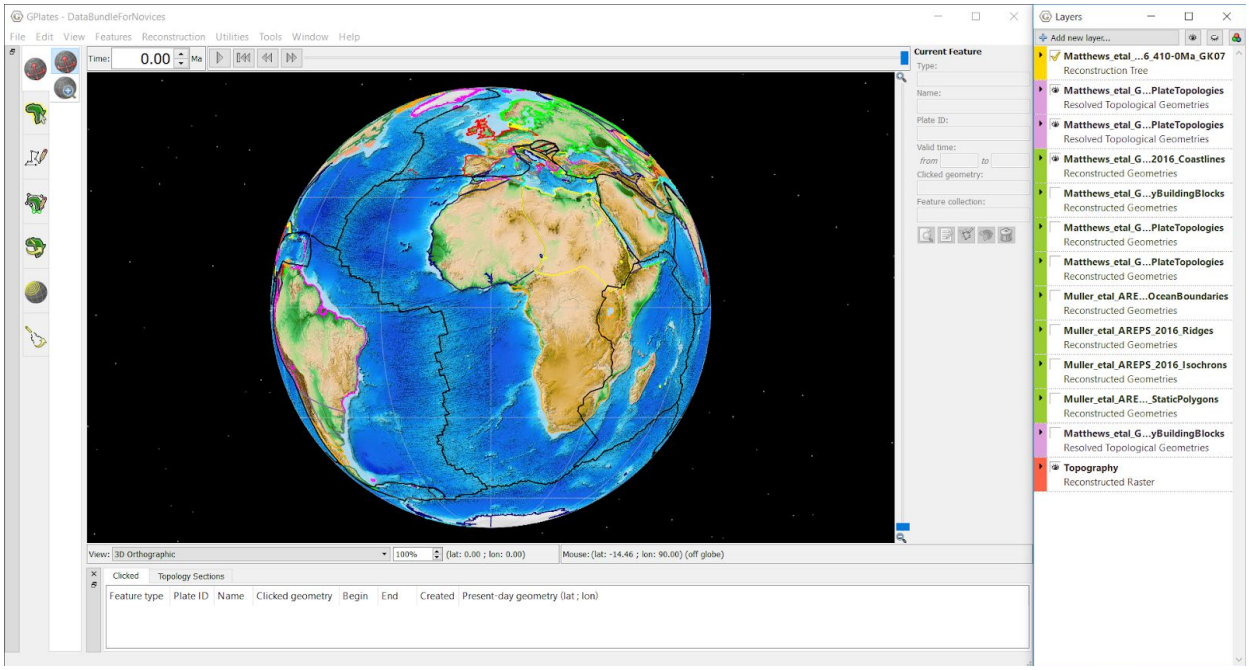

Figure 2. GPlates with all layers except for the resolved topologies, coastlines, and topography disabled.

For our example, we will focus on India as it breaks away from Gondwana from 131 – 122 Ma. Use the time slider to reconstruct back to 131 Ma and drag the globe to the desired location. We can also recolour the coastlines silver using the Manage Colouring tool under the Features menu option (Figure 3).

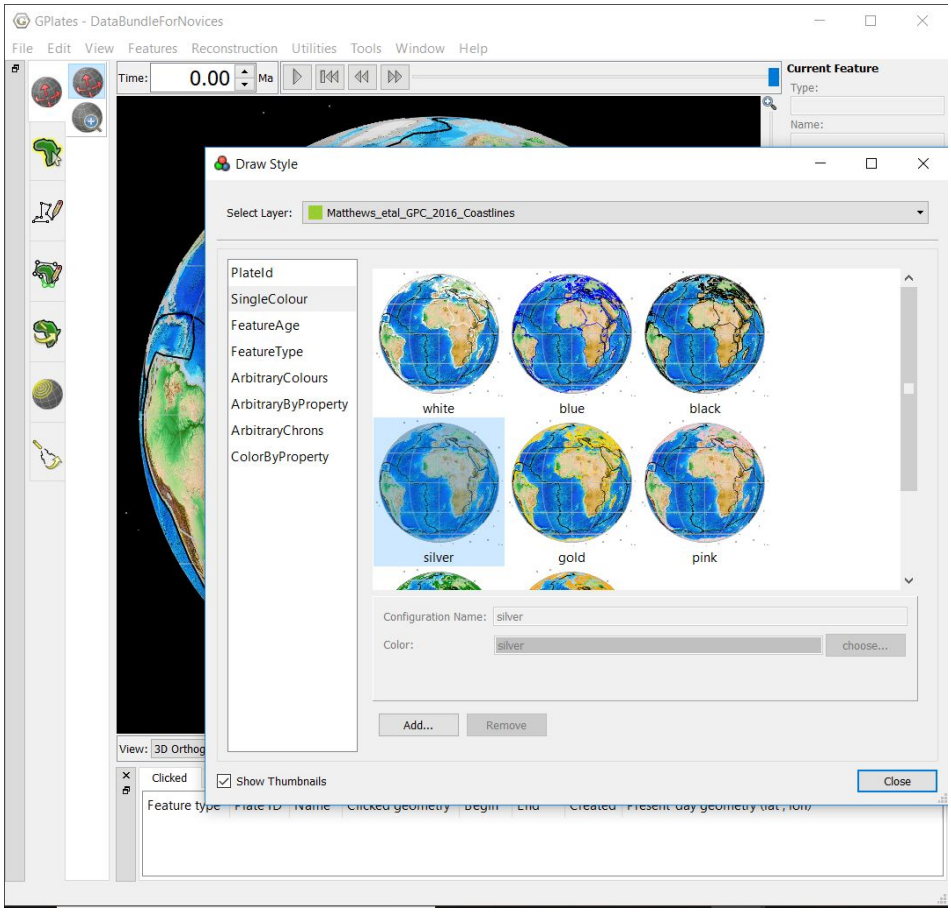

Figure 3. In the Manage Colouring window, select the Coastlines layer and recolour it as silver.

We now need to add a timestamp to the GPlates main window which will be captured by the 'Export' tool. To do this, select 'Configure Text Overlay…' under the View tab (Figure 4). This will open the 'Configure Text Overlay' window.

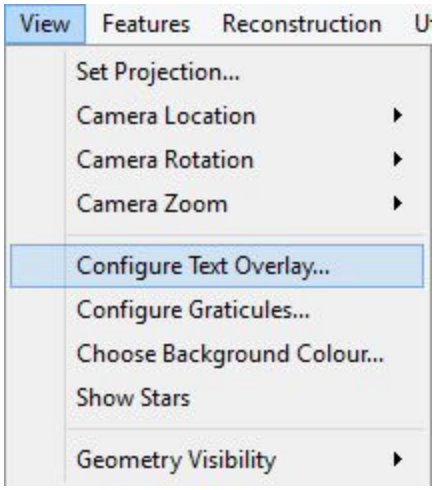

Figure 4. Select 'Configure Text Overlay' to add a timestamp to the GPlates main window.

Check the 'Enable Text Overlay' box to un-grey the interface. The default text to be overlaid is '%f Ma' which represents the current reconstruction time. Leave this as it is.

Other properties which can be specified include the number of decimal places, font, colour, anchor (the position of the text relative to the reconstruction interface), and horizontal and vertical offsets. It is recommended that you make the font size large so that the timestamp is visible from a distance. Once you have finished configuring your text, click OK (Figure 5).

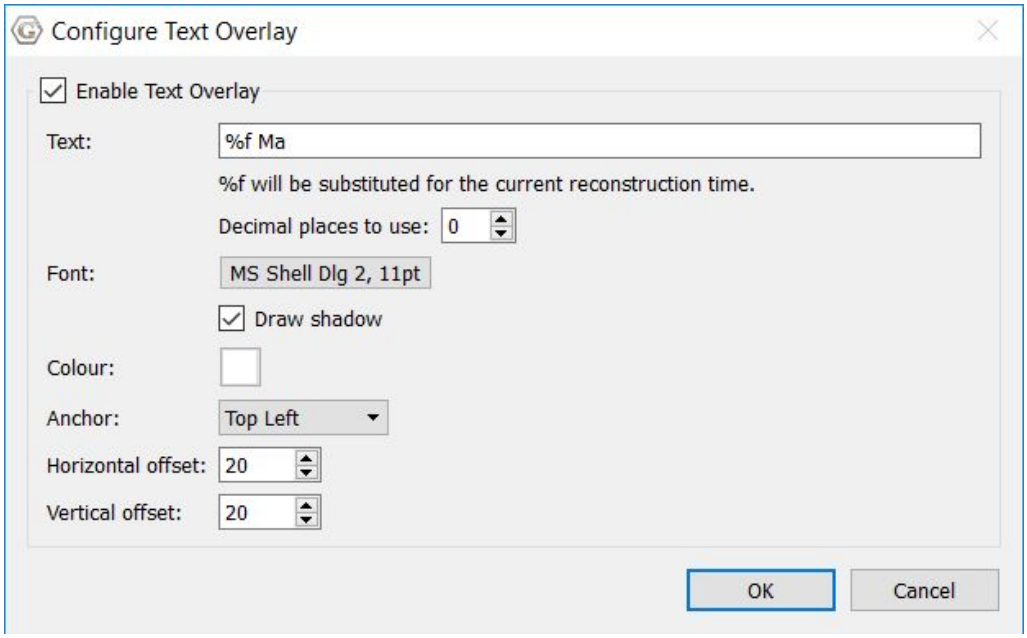

Figure 5. The 'Configure Text Overlay' window where you can specify properties of your timestamp text.

You will now see a timestamp in the location you have specified on the GPlates main window (Figure 6). Try adjusting the reconstruction time using the time slider – you will see that the reconstruction time matches up with the timestamp that we have just inserted.

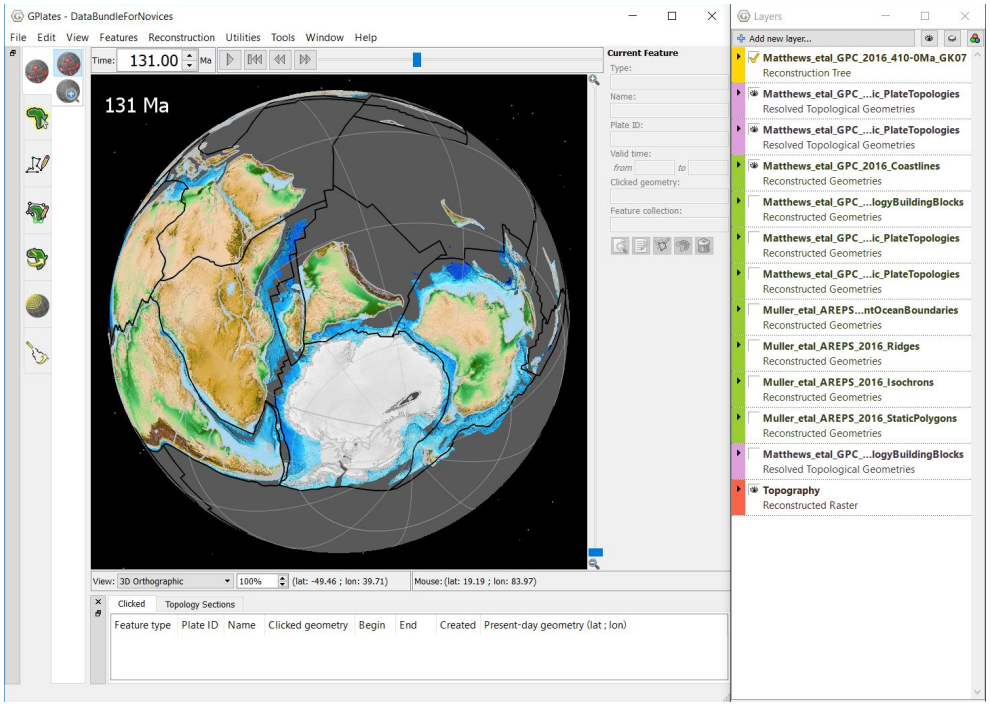

Figure 6. The GPlates main window showing a timestamp of 131 Ma.

Now we will export a series of ten images covering the period 131 – 122 Ma. With the reconstruction time set to 131 Ma and the view centred on India, select the option 'Export…' under the 'Reconstruction' tab (Figure 7). This will open the 'Export' window.

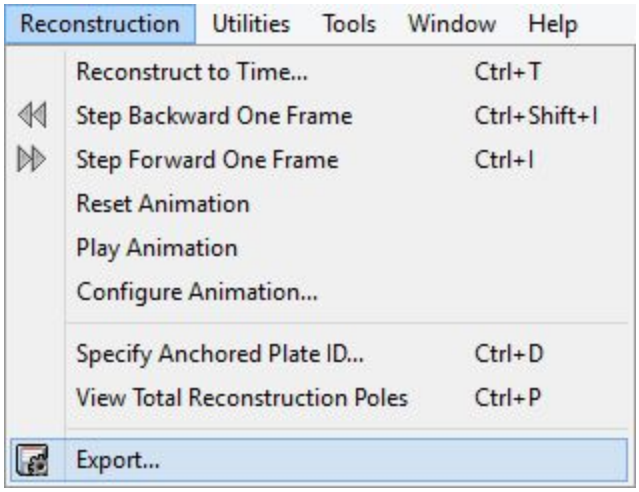

Figure 7. Select the option 'Export...' under the 'Reconstruction' tab.

In the Export window (Figure 8), select 'Export Time Sequence of Snapshots'. Specify the start and end times of the animation, by setting 'Animate from:' to 131 Ma and setting 'to' to 122 Ma, with an increment of 1 Myr per frame. If desired, change the location where the images will be saved by changing the 'Target directory'.

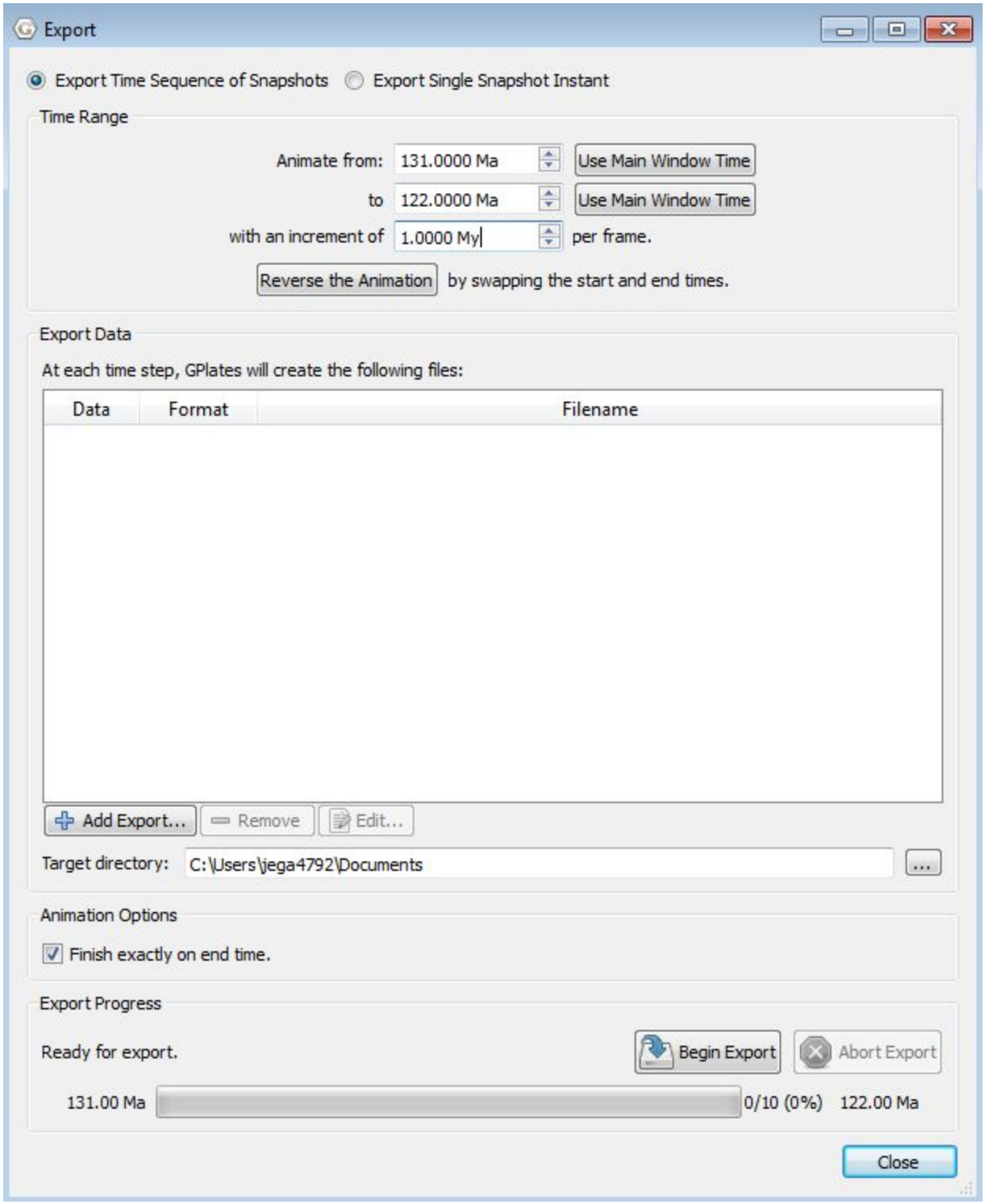
Export
Export Time Sequence of Snapshots
Export Single Snapshot Instant
Time Range
Animate from:
131.0000 Ma
Use Main Window Time
to
122.0000 Ma
Use Main Window Time
with an increment of
1.0000 My
per frame.
Reverse the Animation
by swapping the start and end times.
Export Data
At each time step, GPlates will create the following files:
Data
Format
Filename
Add Export...
Remove
Edit...
Target directory:
C:\Users\jega4792\Documents
...
Animation Options
Finish exactly on end time.
Export Progress
Ready for export.
Begin Export
Abort Export
131.00 Ma
0/10 (0%)
122.00 Ma
Close

Figure 8. Choose the option 'Export Time Series of Snapshots in the Export window and specify the time range you want your images to span.

The next step is to specify what we actually want this snapshot exported as. To do this, click on 'Add Export'. This will open the 'Add Data to Export' window (Figure 9) where you are able to specify the file type, file name, and file size of your image.

Since we are going to export images, select 'Image (screenshot)' under '1. Choose File to Export'. A list of image file types will appear under '2. Choose Output File Format'. Notice that if you experiment by selecting different data types, the types of image files available to you will also change. We will choose to export our images as Joint Photographic Experts Group (.jpg) files; Portable Network Graphics (.png) files also work well.

Once you select this, the '3. Configure Export Options' section will become available. Here you can specify the resolution of the image by selecting the width and height. By default, the width and height are set to the main GPlates window dimensions. If you wish to reset the width and height to this default, simply click the yellow arrow next to 'Use main window dimensions:'. It is recommended that you tick the box for 'Constrain aspect ratio' in order to preserve the width:height ratio of the original reconstruction. Keep in mind that this will grey out the Height option, so if you desire an image with specific dimensions then leave the box unticked.

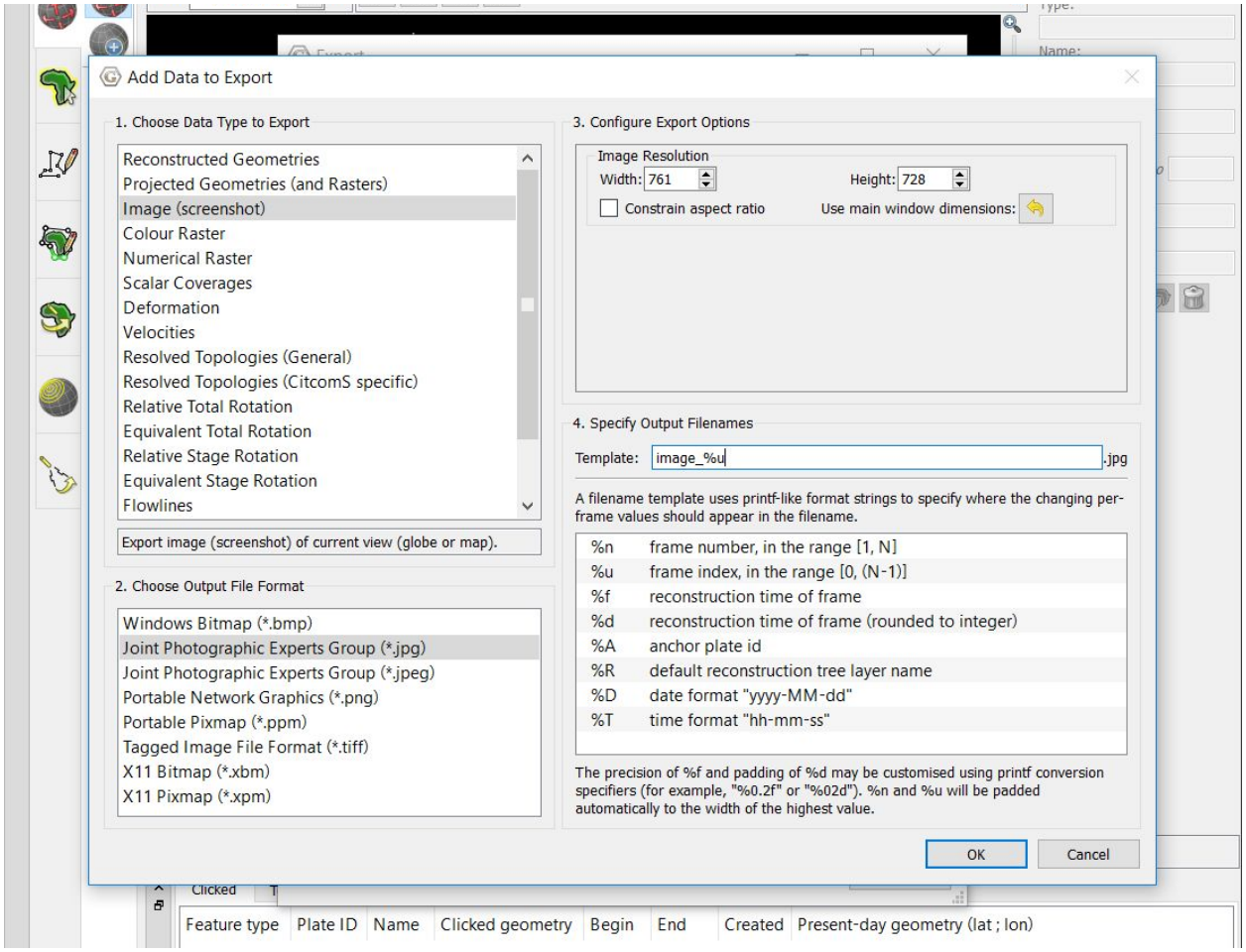

Figure 9. The 'Add Data to Export' window where we can specify the data type, output file format, export size/options and output filename templates for our images of India from 131 – 122 Ma.

The final section, '4. Specify Output File Names', currently contains the default template 'raster_%0.2fMa', which needs to be changed. Our image files need to be named according to the correct formula: each image's filename must end with a 'frame index', starting at 0 with the first image and then increasing sequentially. As can be seen in Figure 9, the template must therefore end with '%u'. Using the template 'image_%u', our image files will be named 'image_00', 'image_01', [...], 'image_09'.

Once we have specified all of these conditions, we can click 'OK'. This will return us to the 'Export' window, which will now list the output conditions we have specified under 'GPlates will create the following files:'. When we click 'Export Snapshot', GPlates will begin exporting the files we have specified, with its progress indicated by the bar at the bottom of the window; this

process may take a long time if a large number of files are being exported. Once GPlates has finished exporting all of the files, they can be found in the target directory we specified earlier.

Exercise 2 – Creating a time-lapse animation from the exported reconstructions

---

FFmpeg is a program which can be used to perform many tasks relating to video files, including creating a video from a series of still images. In this tutorial, we will use FFmpeg to create an animated video from the series of still images which we just exported from GPlates. This process differs depending on the operating system used; the next section is intended for Windows users, and the following section for Mac/Linux users.

Installing and using Avanti and FFmpeg on Windows

In order to use FFmpeg on Windows, we will utilise a program named Avanti. Avanti is a graphical user interface (GUI) for FFmpeg and allows a user to make use of the features of FFmpeg relatively easily, without having to familiarise themselves with FFmpeg's command-line interface (CLI).

Installing Avanti and FFmpeg on Windows

Avanti can be downloaded from here (as of the time of writing, the most recent version is 0.9.2, released 08-04-2015). The downloaded files must be extracted using 7-Zip, which can be downloaded and installed from here. The extracted Avanti files can then be placed in any directory.

Avanti also requires FFmpeg, which can be downloaded here. Choose a recent version and select "Windows 32-bit" and "Static" under the "Architecture" and "Linking" headings, respectively (even if your machine's architecture is 64-bit, Avanti works best with the 32-bit version of FFmpeg). These downloaded files can then be extracted either using 7-Zip or by right-clicking and selecting "Extract All…"

Next, navigate to the "bin" directory within the FFmpeg files and locate the files named "ffmpeg.exe" and "ffplay.exe". Copy these files to the directory named "ffmpeg" within the Avanti files extracted earlier.

Avanti can now be launched by running the file named "Avanti-GUI.exe" within the "Avanti-ffmpeg-GUI-XXX" folder (replace "XXX" with the

appropriate Avanti version number). Avanti may take a few moments to start, as indicated by a message at the bottom of the window: "Please wait for system ready!"

Additionally, it may sometimes be necessary to run Avanti in 'Compatibility Mode' – this is done by: right-clicking on Avanti-GUI.exe → selecting "Properties" → selecting the "Compatibility" tab → checking the box marked "Run this program in compatibility mode for:" → selecting "Windows XP (Service Pack 3)" from the drop-down menu.

Using Avanti on Windows

Once we have created our series of image files and installed the required software, we can use Avanti to stitch the images together into an animated video file. Launch Avanti as described above, in the "Installing Avanti and FFmpeg on Windows" section of this tutorial.

Once Avanti is open, we must first specify that our input is a sequence of images, by right-clicking on the text box labelled "Source 1" and selecting "Image sequence" (Figure 10).

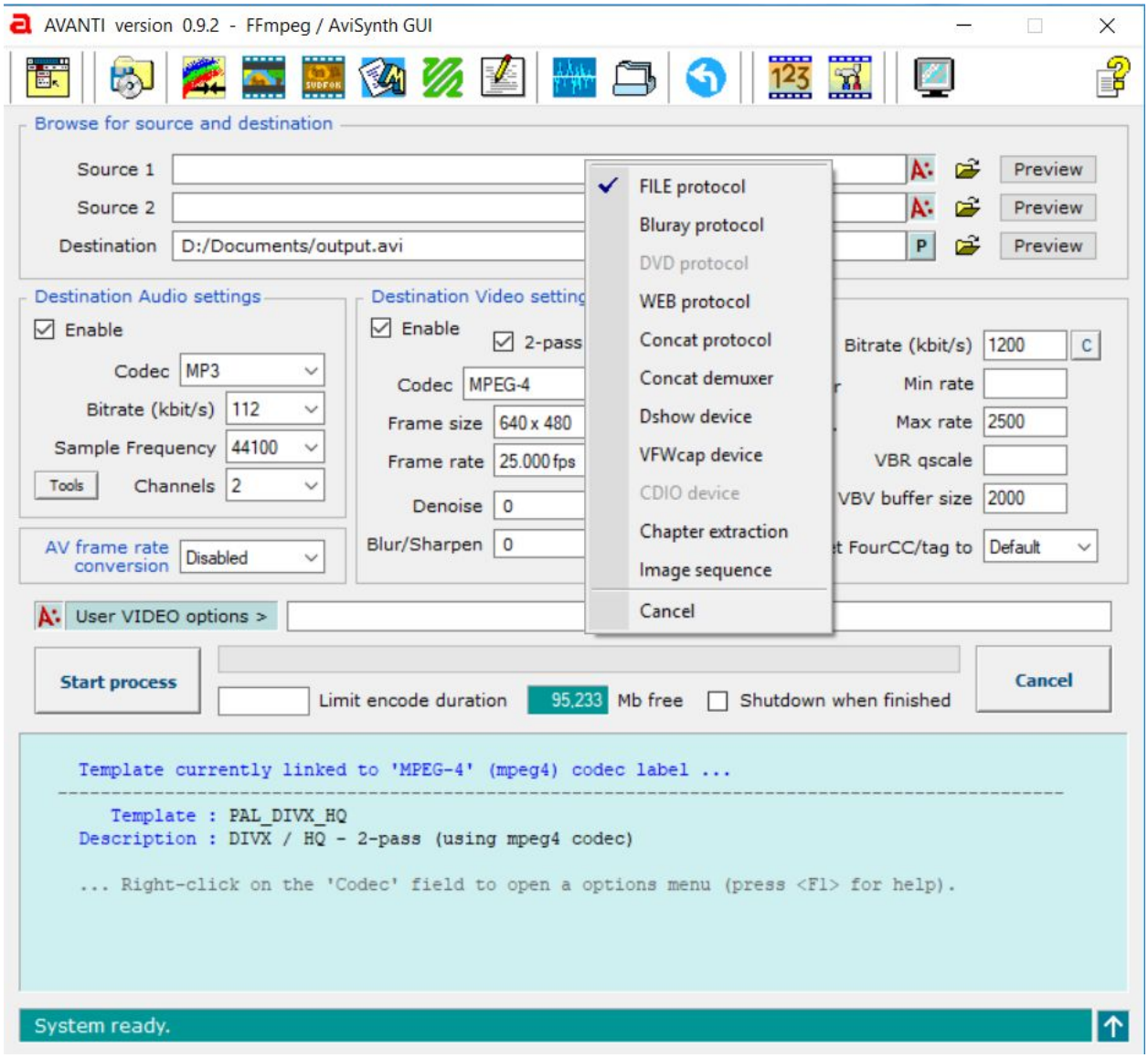

Figure 10. Choose the "Image sequence" option for the "Source 1" field.

Then, we click on the folder icon next to the box marked "Source 1", locate the image files we produced earlier, and double-click on one of them (Figure 11). Avanti should recognise the naming scheme for our files, as can be seen in Figure 11: "image_%02d.jpg", where "%02d" indicates a sequence of two-digit numbers, in this case running from "00" to "09".

Similarly, we can choose a destination for our video output using the folder icon next to the box marked "Destination".

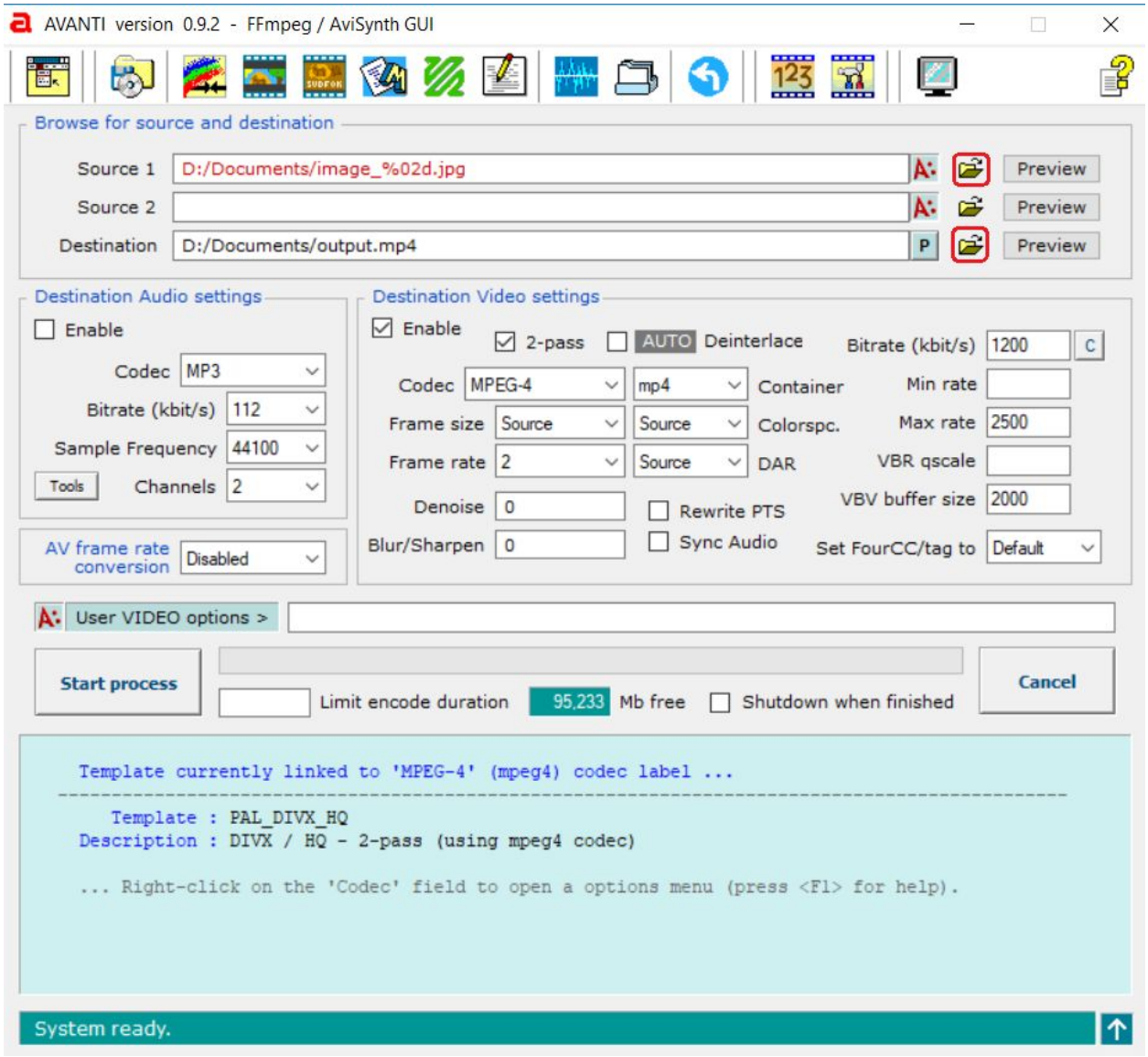

Figure 11. Use the folder icons to select the input and output file locations.

Avanti offers many options for video manipulation, only some of which will be useful to use. Firstly, we need to make sure that the "Destination Audio settings" are disabled (Figure 12) — since our input image files have no audio, neither will our output video file.

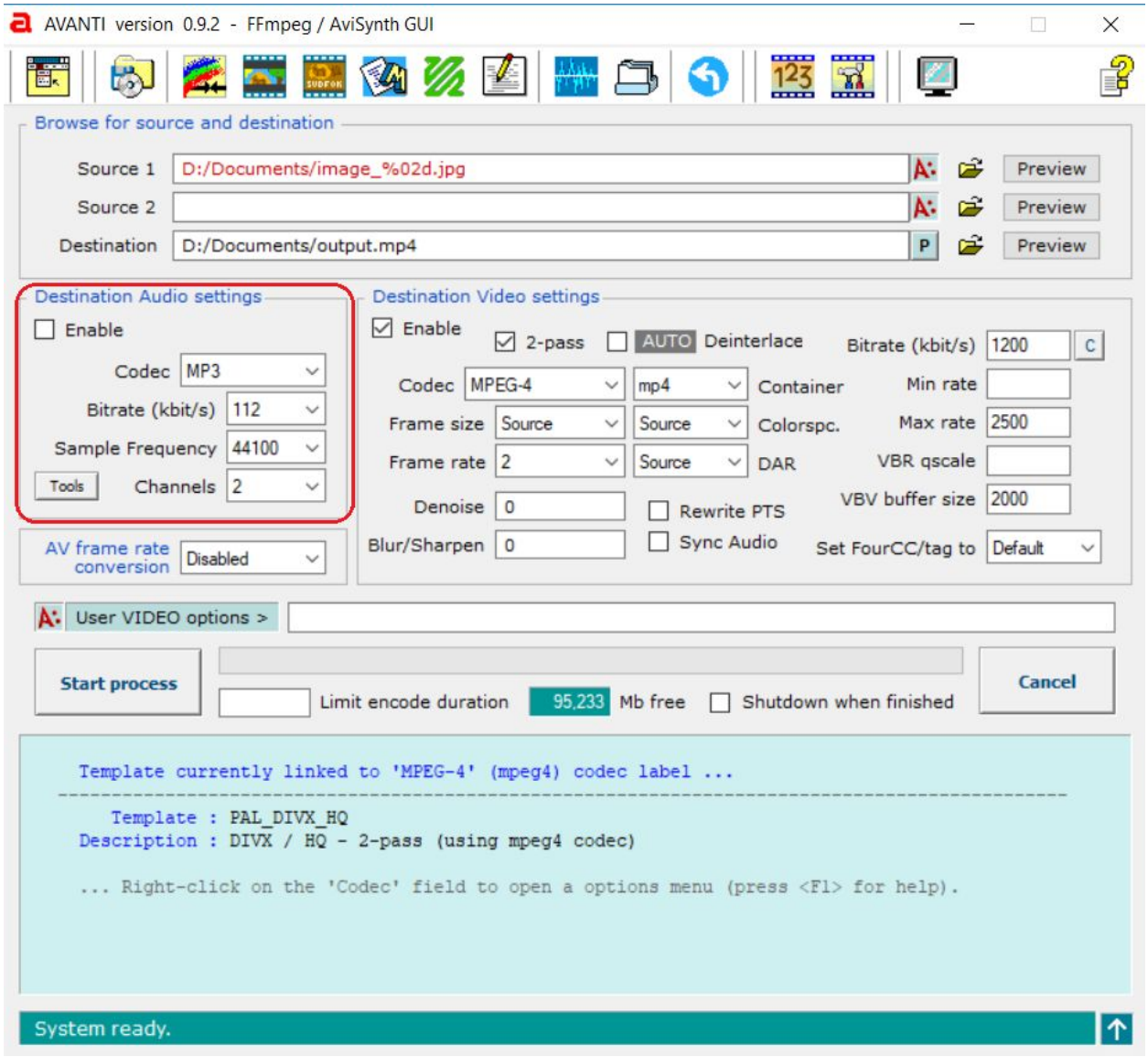

Figure 12. Uncheck the box marked "Enable" under "Destination Audio settings".

Next (Figure 13), we can choose the output video file's codec (MPEG-4 usually works well) and container (file extension, such as .mp4 or .avi).

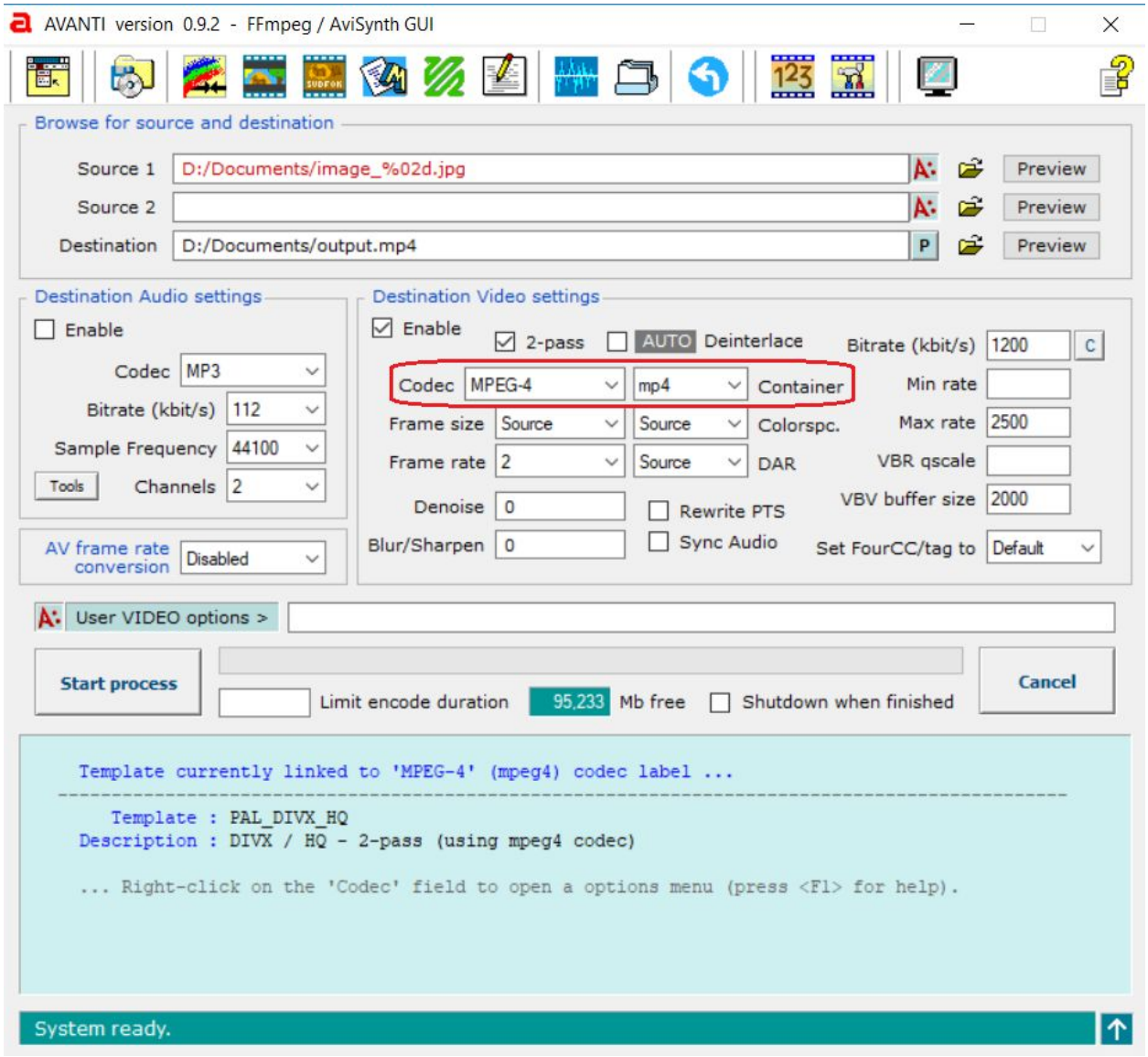

Figure 13. Select the codec and container for the video output.

If the "Frame size" option (Figure 6) is set to "Source", the resolution of the video will be the same as that of the images; if different dimensions are needed, a different resolution (in pixels, e.g. "600 x 600") can be entered.

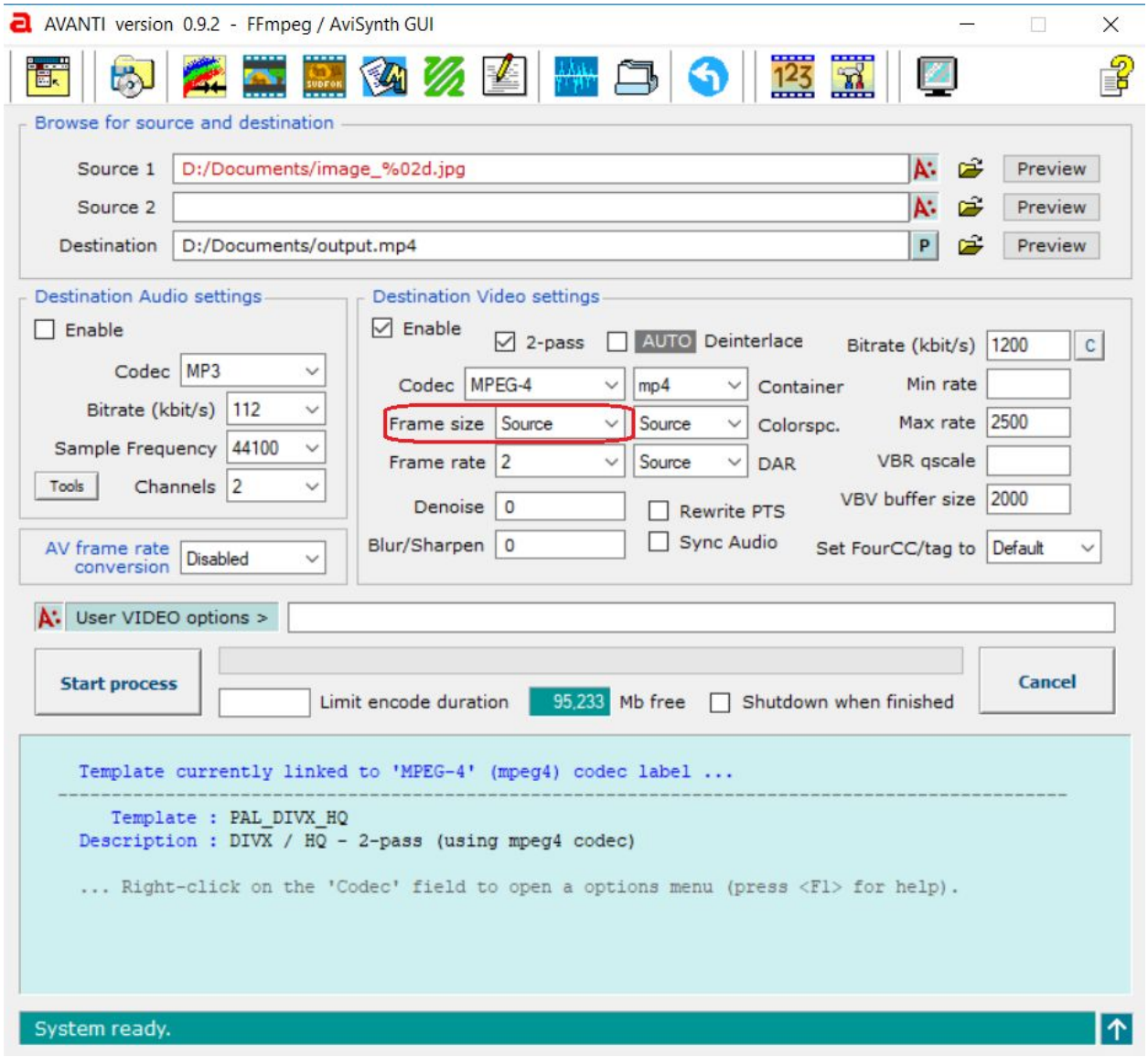

Figure 14. Leave the "Frame size" parameter on "Source" or change it to the desired video resolution.

The "Frame rate" setting (Figure 15) is important: this controls how quickly the images will change in the animated video. It is measured in frames per second, so that a frame rate of 1 is equivalent to 1 second per image, 2 is equivalent to 0.5 seconds per image, etc.

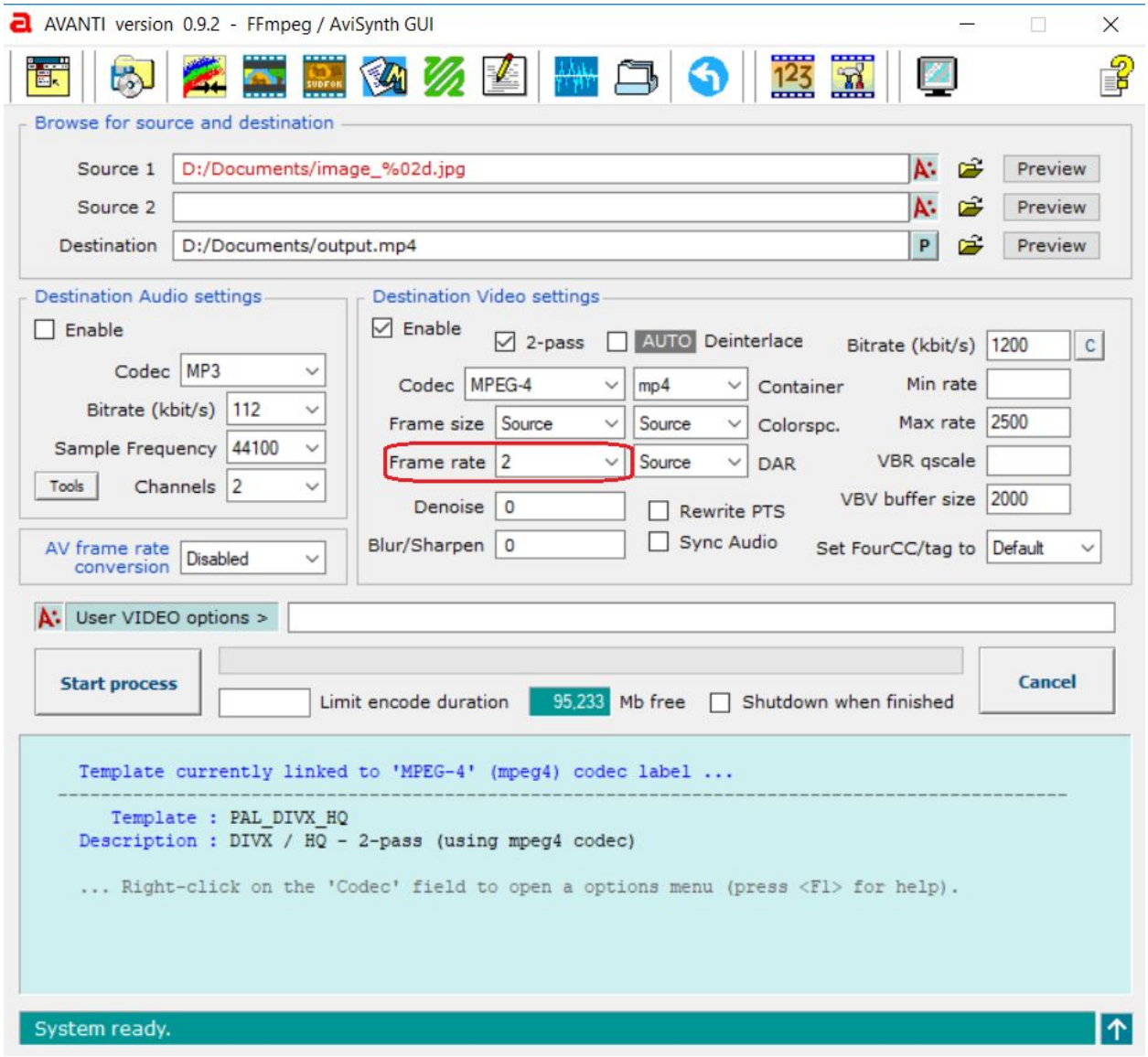

Figure 15. Enter the desired frame rate for the animated video. In this example, a frame rate of 2 equates to 0.5 seconds per image.

The "Bitrate" setting (Figure 16) determines the quality (and therefore the file size) of the video. If a smaller file size is needed and a high quality is not crucial, this number can be lowered.

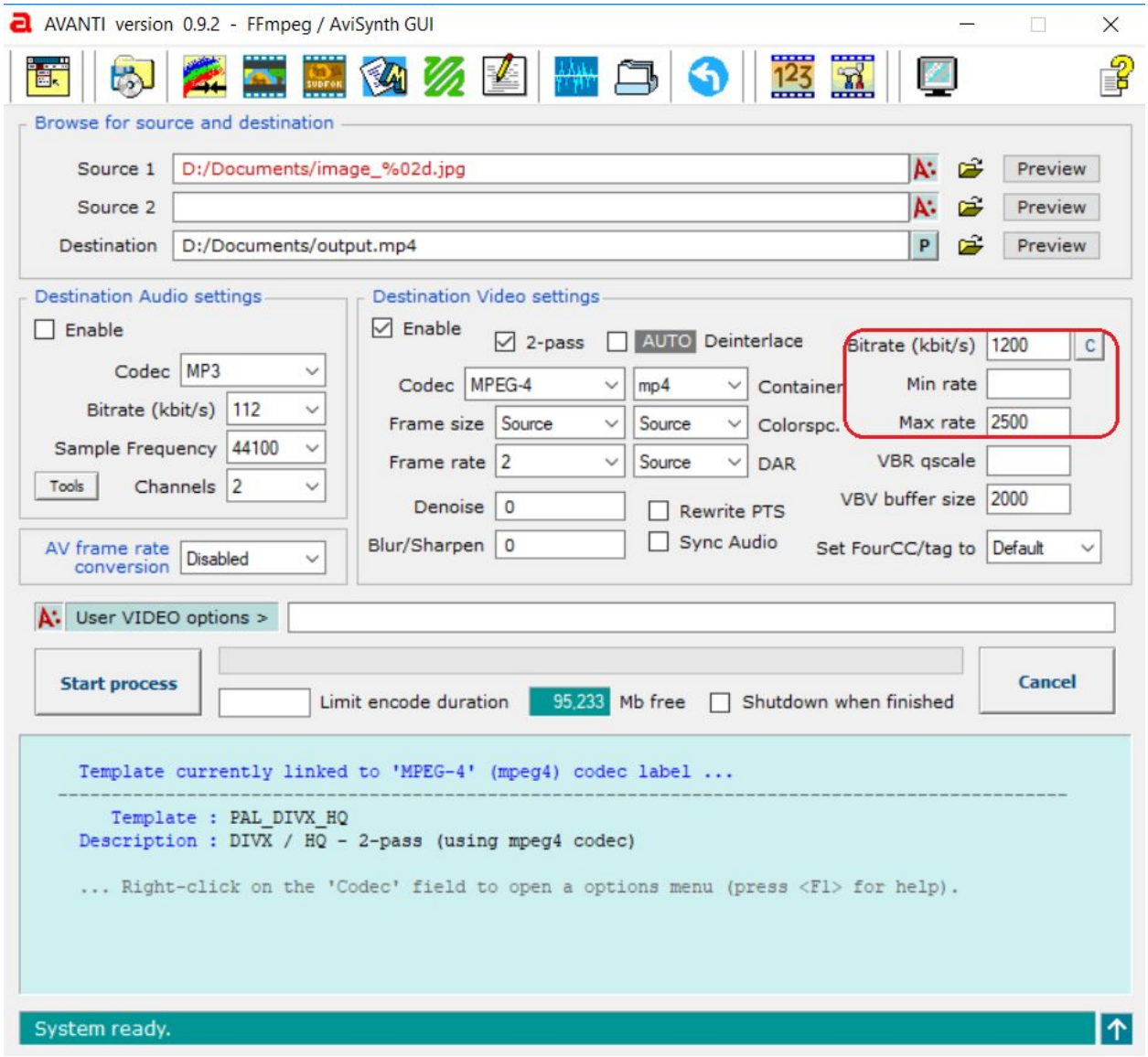

Figure 16. The bitrate (measured in kbit/s) of the video can be adjusted if necessary.

The other available options should mostly be left as "Source" or "Default". If a video aspect ratio (width:height ratio) different to that of the input images is required, this can be changed using the "DAR" option.

Finally, we can click the "Start process" button to begin creating our video (Figure 17).

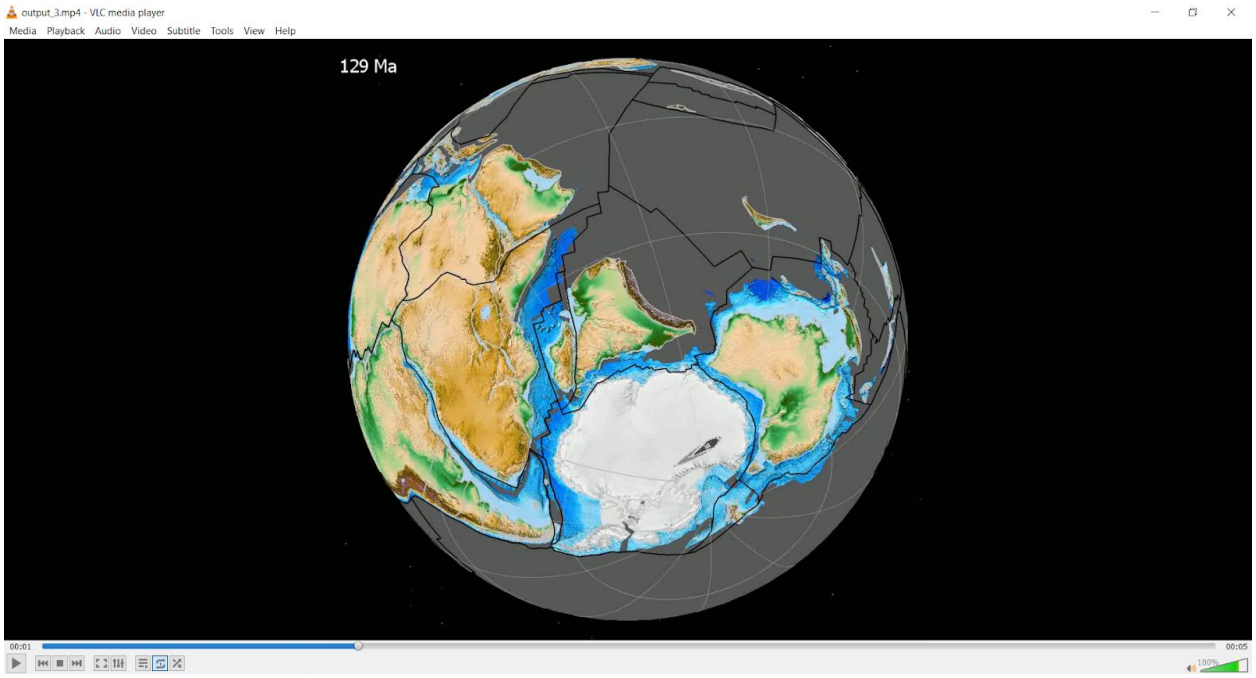

Figure 17. Success!

Installing and using FFmpeg on Mac/Linux

Installing FFmpeg is significantly less complicated on Mac and Linux than on Windows, so we will bypass Avanti and install and use FFmpeg directly from the operating system's command line instead.

Installing FFmpeg on Mac/Linux

Simply use your preferred package manager to install FFmpeg. For example, if using Macports, open a Terminal window and enter "sudo port selfupdate", followed by "sudo port install ffmpeg".

Similarly, on Ubuntu, open a Terminal window and enter "sudo apt update", followed by "sudo apt install ffmpeg".

Using FFmpeg on Mac/Linux

Once FFmpeg has been installed, use the Terminal window's "cd" command to navigate to the directory containing the image files exported earlier.

FFmpeg is a powerful tool which provides many options (see the documentation here). Rather than familiarising yourself with FFmpeg's extensive syntax, you can simply copy and paste the following line, modifying it to your specifications:

ffmpeg -framerate 2 -pattern_type sequence -i 'image_%02d.jpg' -vf scale=800:-1 -vcodec mpeg4 -b 5000k -r 2 output.mp4

The different elements of the above command are as follows:

-framerate	Sets the framerate of the output video file (in frames per second); "-framerate 2" is 2 frames per second, or 0.5 seconds per image
-pattern_type	"-pattern_type sequence" specifies that the input is a sequence of image files
-i	Specifies the input file(s); "image_%02d.jpg" indicates a series of file names beginning with "image_" and followed by a two-digit number beginning at "00"
-vf	"-vf scale=800:-1" is optional, and resizes the output video to a width of 800 pixels and a corresponding height (keeping the same aspect ratio (width:height))
-vcodec	Specifies the codec of the output video file; "-vcodec mpeg4" indicates that the output will be encoded using MPEG-4
-b	Specifies the bitrate of the output video file; "-b 5000k" indicates a bitrate of 5000 kbit/s. Increase for higher quality but larger file sizes
-r	Should be the same as -framerate; otherwise, frames might be dropped from the output video file

output.mp4	Specifies the name of the output video file. For a different file extension, change ".mp4" to ".avi", ".mov", etc.