G- Streamer basics :

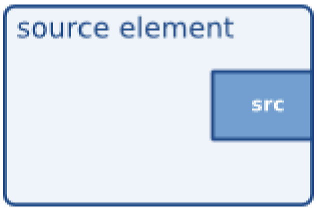
To execute any GStreamer pipeline commands ,mainly we will require 3 Gstreamer elements
1. Source

2. Filters, convertors, demuxers, muxers and decoders

3. Sink

1) Source element

Source elements generate data for use by a pipeline, for example reading from disk or from a sound card. Visualization of a source element shows how we will visualize a source element. We always draw a source pad to the right of the element.

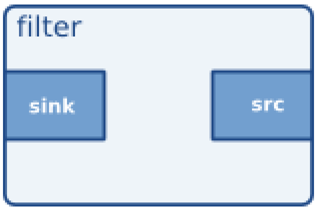


Source elements do not accept data, they only generate data. You can see this in the figure because it only has a source pad (on the right). A source pad can only generate data.

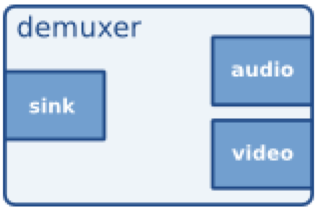
2) Filters, convertors, demuxers, muxers and decodrcs

Filters and filter-like elements have both input and outputs pads. They operate on data that they receive on their input (sink) pads and will provide data on their output (source) pads. Examples of such elements are a volume element (filter), a video scaler (convertor), an qtdemux (demuxer) or a v4l2h264dec (decoder).

Filter-like elements can have any number of source or sink pads. A video demuxer, for example, would have one sink pad and several (1-N) source pads, one for each elementary stream contained in the container format. Decoders, on the other hand, will only have one source and sink pads.



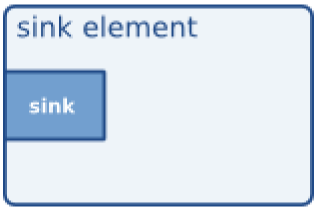
Visualization of a filter element shows how we will visualize a filter-like element. This specific element has one source pad and one sink pad. Sink pads, receiving input data, are depicted at the left of the element; source pads are still on the right.



Visualization of a filter element with more than one output pad shows another filter-like element, this one having more than one output (source) pad. An example of one such element could, for example, be an (qtdemx) demuxer for an qtdemux stream containing both audio and video. One source pad will contain the elementary video stream, another will contain the elementary audio stream. Demuxers will generally fire signals when a new pad is created. The application programmer can then handle the new elementary stream in the signal handler.

3) Sink elements

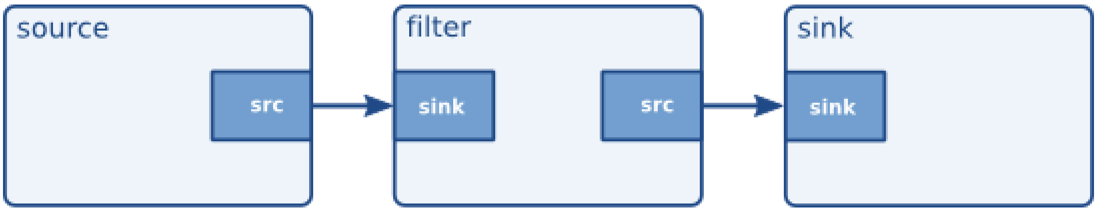
Sink elements are end points in a media pipeline. They accept data but do not produce anything. Disk writing, soundcard playback, and video output would all be implemented by sink elements. Visualisation of a sink element shows a sink element.



Displaying a test pattern generated by G-Streamer :

1----->In simple form, a PIPELINE-DESCRIPTION is a list of element types separated by exclamation marks (!). Go ahead and type in the following command:

```
gst-launch-1.0 videotestsrc ! videoconvert ! autovideosink
```



You should see a windows with an animated video pattern. Use CTRL+C on the terminal to stop the program.

This instantiates a new element of type `videotestsrc` (an element which generates a sample video pattern), an `videoconvert` (an element which does raw video format conversion, making sure other elements can understand each other), and an `autovideosink` (a window to which video is rendered). Then, GStreamer tries to link the output of each element to the input of the element appearing on its right in the description. If more than one input or output Pad is available, the Pad Caps are used to find two compatible Pads.