



Datamodel to playback graph

Arrangement

• ~2000 lines of code in Traction Engine

• Not a widely standard forward

• Aux send/returns are just numbers and positions in the model

• Graph needs to create nodes with shared buffers, gaining, summing nodes which handle ordering and latency directly

• Track minutes/sold



• Two properties control whether a track should be muted or not

- Including upstream/downstream tracks from folders/group

• Lots of people think the graph needs to be optimised



• Clips don't all need to be pressed all the time, only the ones that play the

• Mutated/bypassed plugins may need to be processed if they introduce latency/unmutated/bypassed correctly

**D: 835**

**£251k**

# Arrangement

## Data model to playback graph

**D: 835    £251k**

- ~2000 lines of code in Tracktion Engine
- Not always straightforward
  - Aux send/returns are just bus numbers and positions in the model
    - Graph needs to create nodes with shared buffers, gains, summing nodes which all handle ordering and latency correctly
- Track mute/solo
  - Two properties control whether a track should be muted or not
  - Including upstream/downstream tracks from folders/groups
- Lots of places the graph needs to be optimised
  - Clips on a track don't all need to be processed all the time, only those near the playhead
  - Muted tracks/bypassed plugins *may* need to be processed if they introduce latency in order to un-mute/un-bypass correctly

# Audio Recording