

Performance format description

This document describes the format of the Electra One Performance file. A Performance is a special feature that allows users to create an additional page with a customized selection of controls from an associated preset. Its main purpose is to make it easier for users to gather controls that may be spread across multiple preset pages into a single, easy-to-access page. It also enables the creation of macro controls, modulation of control values, and other performance related tools.

Preset controls are linked to the Performance page using references to preset controls and control values. References can be either:

- Simple references, which point to a single control, or
- Multi-references, which link to multiple control values at once.

Multi-references are commonly used to build macro controls or to apply modulation using Electra One modulation sources (data pipes).

A Performance must always be associated with a preset, it cannot function independently. This association is established by placing the Performance file into the same preset slot as the preset file, and by referencing the preset's controls through their control and value Ids.

The Performance feature was introduced in firmware version 4.0. As it is a relatively new addition, its functionality is expected to continue evolving and expanding over time.

Performance JSON format

JSON schema

The JSON schema of the Electra One Performance file is available at [GitHub](#).

Top level objects

The performance has three top-level elements.

```
{
  "version": 1,
  "references": [
  ],
  "groups": [
  ]
}
```

version

Provides information about the version of the performance file. Electra One controller uses version information to distinguish between various performance file formats. Note, this document describes performance version 1.

- mandatory
- numeric

example

```
"version":1
```

references

An array of references. Each reference defines a control that will appear on the Performance page. Its position on the page is determined by the `controlSetId` and `potId`. The control being referenced is specified either by a `controlId` (for a simple reference) or a `valueRefs` array (for a multi-reference).

- mandatory
- array

example

```
"references": [
  {
    "controlSetId":1,
    "potId":1,
    "controlId":1
  },
  {
    "controlSetId":1,
    "potId":6,
    "valueRefs": [
      {
        "controlId":1,
        "valueId":"value",
        "mode":"dataPipe",
        "pipe":{
          "name":"output",
          "bankNumber":5,
          "slot":1
        }
      },
      {
        "controlId":2,
        "valueId":"value",
        "mode":"setValue"
      },
      {
        "controlId":3,
        "valueId":"value",
        "mode":"modulate",
        "depth":50
      }
    ]
  },
  {
    "name":"ALL FADERS"
```

```
}
]
```

groups

An array of groups. A group is a graphical separator to improve a layout of presets. The performance groups are completely independent from preset groups. On contrary to preset, the performance groups do not have `id` and `pageId` attributes.

- optional
- array

example:

```
"groups": [
  {
    "name": "ATTRIBUTES",
    "bounds": [
      170,
      16,
      485,
      16
    ],
    "color": "FFFFFF"
  }
]
```

Reference

A reference links controls from the associated preset to the Performance page. It can either point to a single control (simple reference) or to multiple controls at once (multi-reference). References allow you to reuse and arrange existing controls without modifying the original preset, enabling custom layouts, macro controls, or modulation routing.

A simple-reference links one preset control using its `controlId` and assigns it to a specific control set and pot (knob) on the performance page.

controlId

An identifier of the control to be referenced at the specified control set and pot position. The `controlId` must correspond to a control from the preset associated with the Performance page.

When `controlId` is used, the reference is a simple reference that links to a single control from the preset. To link multiple control values, a `valueRefs` array must be used instead (see below).

- optional (either `controlId` or `valueRefs`)
- numeric
- min = 1
- max = 1023

controlSetId

Controls references placed on the performance page can be further organized into control sets, three rows of twelve controls. Control sets are used to assign controls to the physical pots (knobs). Users can switch between control sets by pressing hardware buttons, using the touchscreen, or by sending MIDI messages to Electra.

Only one control set can be active at a time. The controls belonging to the active control set are highlighted by indicators on the sides of the screen.

- mandatory
- numeric
- min = 1
- max 3

potId

An identifier of the physical pot (knob). Electra One has twelve pots, numbered from 1 (top-left) to 12 (bottom-right). When a control is assigned to a pot, it can be adjusted by turning the corresponding physical knob, provided in its in the current control set.

- mandatory
- numeric
- min = 1
- max = 12

example:

```
{
  "controlSetId":1,
  "potId":1,
  "controlId":1
}
```

valueRefs

An array of references to control values in the associated preset. Each control value is identified by a combination of `controlId` and `valueId`. The array forms a multi-reference that links multiple control values to a single control on the Performance page. This allows to create macro controls that adjust several parameters at once or to apply modulation to controls.

- optional (either `controlId` or `valueRefs`)
- array

When `valueRefs` is used, the reference must include a name attribute, since the `name` cannot be automatically determined from the control values.

example:

```

{
  "controlSetId":1,
  "potId":6,
  "valueRefs":[
    {
      "controlId":1,
      "valueId":"value",
      "mode":"dataPipe",
      "pipe":{
        "name":"output",
        "bankNumber":5,
        "slot":1
      }
    },
    {
      "controlId":2,
      "valueId":"value",
      "mode":"setValue"
    },
    {
      "controlId":3,
      "valueId":"value",
      "mode":"modulate",
      "depth":50
    }
  ],
  "name":"ALL FADERS"
}

```

name

Name of the control reference. It can be used only in combination with multi-references using the `valueRefs` .

- optional
- string
- default = ""
- minLength = 0
- maxLength = 14

Value References

A value reference can link up to 16 preset control values to single control reference, and thus they can be controlled with one pot (knob) or a modulation source.

valueId

An identifier of the value within the "values" array of the referenced control.

- mandatory
- string

- default = value
- minLength = 1
- maxLength = 20

There are three ways of linking the control values to the common reference, they are referred as to **mode**. The available modes are:

- setValue - simple value mapping
- modulate - value modulation
- dataPipe - value modulation using the data pipe

mode

Defines a mode how the control value is linked to the control reference. Each mode may require additional parameters.

- mandatory
- enum
 - "setValue"
 - "modulate"
 - "dataPipe"

setValue

The setValue mode maps control value range to full range (0 .. 127) of the control reference. The setValue mode does not require any additional parameters.

```
{
  "controlId":2,
  "valueId":"value",
  "mode":"setValue"
}
```

modulate

The modulate mode applies a modulation value to the current value of the referenced control value. This allows users to control both the base value of the parameter as well as amount of modulation applied to it. This principle is similar to how macro knobs work on Elektron devices.

The modulate mode requires a depth parameter to be specified.

```
{
  "controlId":3,
  "valueId":"value",
  "mode":"modulate",
```

```
"depth":50
}
```

dataPipe

The dataPipe mode is very similar to modulation mode described above.

```
{
  "controlId":1,
  "valueId":"value",
  "mode":"dataPipe",
  "pipe":{
    "name":"output",
    "bankNumber":5,
    "slot":1
  }
}
```

pipe

The pipe parameter is used to specify source of the data pipe data. It is a JSON object with three attributes:

- mandatory for dataPipe mode
- object

bankNumber

A bank number where the preset that generates data pipe data is stored.

- mandatory
- numeric
- min = 0
- max 5

slot

A slot where the preset that generates data pipe data is stored.

- mandatory
- numeric
- min = 0
- max 11

name

Name of the pipe output as defined in the preset that generates data pipe data.

- mandatory
- string
- minLength = 0
- maxLength = 14

Group

Graphical separators used to organize control references into meaningful groups. For example, you could create a group called "Envelope 1" that contains the controls "Attack," "Decay," "Sustain," and "Release." Groups serve only a visual purpose and do not add any additional functionality beyond organizing controls for easier navigation.

example:

```
{
  "name": "ENVELOPE",
  "bounds": [
    0,
    16,
    486,
    16
  ],
  "color": "FFFFFF",
  "variant": "highlighted"
}
```

name

The name of the group. It is displayed to the user inside the group's graphical area and is automatically trimmed to fit the available space.

- mandatory
- string
- minLength = 0
- maxLength = 40

bounds

The bounding box of the control, defining its position and size on the screen. It is represented as an array in the format: [x, y, width, height].

- mandatory
- array with fixed items

color

A 24-bit RGB code defining the group's color. Electra One internally uses 16-bit RGB565 color format, so the final displayed color may differ slightly due to conversion.

- optional
- string
- default = FFFFFFFF
- minLength = 6
- maxLength = 6

variant

The variant of the group, which determines its visual style.

- optional
- enum
 - "default"
 - "highlighted"
- default = "default"

Example of Performance JSON

For reference, we provide a full example of the Performance file.

example

```
{
  "version":1,
  "references":[
    {
      "controlSetId":1,
      "potId":1,
      "controlId":1
    },
    {
      "controlSetId":1,
      "potId":6,
      "valueRefs":[
        {
          "controlId":1,
          "valueId":"value",
          "mode":"dataPipe",
          "pipe":{
            "name":"output",
            "bankNumber":5,
            "slot":1
          }
        }
      ]
    }
  ],
}
```

```
{
  {
    "controlId":2,
    "valueId":"value",
    "mode":"setValue"
  },
  {
    "controlId":3,
    "valueId":"value",
    "mode":"modulate",
    "depth":50
  }
],
"name":"ALL FADERS"
}
],
"groups":[
  {
    "id":4,
    "pageId":1,
    "name":"GROUP LABEL",
    "color":"ffffff",
    "bounds":[
      14,
      6,
      993,
      171
    ]
  }
]
}
```