

Preset format description

This document describes the format of the Electra One preset file. The preset file holds a complete definition of Electra One preset, no other data needs to be transferred to Electra to run a preset.

Preset JSON format

JSON schema

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

Minification

We strongly advise minifying the JSON data prior to transfer to Electra One over the USB MIDI protocol. Minification greatly affects the amount of data that must be transferred, therefore it speeds up the time of uploading and downloading presets.

Top level objects

The preset has a number of top-level objects. These are either simple elements providing info about the preset itself or complex objects that define the structure and data of the preset.

```
{
  "version": 2,
  "name": "ADSR Test",
  "projectId": "d8WjdwYrP3lRyyx8nEMF",
  "pages": [
  ],
  "devices": [
  ],
  "overlays": [
  ],
  "groups": [
  ],
  "controls": [
  ]
}
```

version

Provides information about the version of the preset file. Electra controller uses version information to distinguish between various preset file formats.

- mandatory
- numeric

name

A name of the preset. The name will be shown to the user on the screen in the status bar.

- mandatory
- string
- minLength = 0
- maxLength = 20

projectId

An external identifier. This id can be used to refer any external data within your Electra applications. For example, the projectId is used to associate a preset within the meta data, such as the preset description, author, etc., in the Electra Preset Editor application.

- optional
- string
- minLength = 0
- maxLength = 20

pages

An array of pages. A page is a collection of controls and groups displayed to the user at once.

- mandatory
- array

example:

```
"pages": [  
  {  
    "id": 1,  
    "name": "OSCILLATORS"  
  },  
  {  
    "id": 2,  
    "name": "FILTER"  
  }  
]
```

devices

An array of devices. A device is a hardware MIDI device or software device (application, VST/AU plugin) connected to the Electra One.

- mandatory
- array

example:

```

"devices": [
  {
    "id": 1,
    "name": "My MKS-50",
    "instrumentId": "roland-mks50",
    "port": 1,
    "channel": 1,
    "rate": 20
  },
  {
    "id": 2,
    "name": "BeatFX plugin",
    "instrumentId": "generic-MIDI",
    "port": 2,
    "channel": 1
  }
]

```

overlays

An array of overlays. An overlay is a list of text labels or graphical symbols that can be assigned to a control.

- optional
- array

examples:

```

"overlays": [
  {
    "id": 1,
    "items": [
      {
        "value": 0,
        "label": "SAW"
      },
      {
        "value": 1,
        "label": "SQUARE"
      },
      {
        "value": 2,
        "label": "WHITE NOISE"
      },
      {
        "value": 4,
        "label": "PINK NOISE"
      }
    ]
  },
  {
    "id": 2,
    "items": [

```

[illegible]

groups

An array of groups. A group is a graphical separator to improve a layout of presets.

- optional
- array

example:

```
"groups": [
  {
    "pageId": 1,
    "name": "ATTRIBUTES",
    "bounds": [
      170,
      16,
      485,
```

```

        16
      ],
      "color": "FFFFFF"
    }
  ]

```

controls

An array of controls. A control is a representation of one or more MIDI parameters or messages that can be controlled by the user.

- mandatory
- array

examples:

A simple control with one value assigned.

```

"controls": [
  {
    "id": 1,
    "type": "fader",
    "name": "WHITE",
    "color": "FFFFFF",
    "bounds": [
      0,
      40,
      146,
      56
    ],
    "pageId": 1,
    "controlSetId": 1,
    "inputs": [
      {
        "potId": 1,
        "valueId": "value"
      }
    ],
    "values": [
      {
        "id": "value",
        "message": {
          "deviceId": 1,
          "type": "cc7",
          "parameterNumber": 1,
          "min": 0,
          "max": 127
        },
        "min": 0,
        "max": 127
      }
    ]
  }
],
{

```

```

    "id": 2,
    "type": "fader",
    "name": "RED",
    "color": "F45C51",
    "bounds": [
      170,
      40,
      146,
      56
    ],
    "pageId": 1,
    "controlSetId": 1,
    "visible": false,
    "inputs": [
      {
        "potId": 2,
        "valueId": "value"
      }
    ],
    "values": [
      {
        "message": {
          "deviceId": 1,
          "type": "cc7",
          "parameterNumber": 2,
          "min": 0,
          "max": 127
        },
        "min": 0,
        "max": 127,
        "formatter": "formatFractions"
      }
    ]
  }
}
]

```

Page

A page is a collection of controls and graphic objects shown on the screen at once. Each page has a name to make it easier to switch between pages. The page name is shown in the status bar.

example:

```

{
  "id": 1,
  "name": "OSCILLATORS"
}

```

id

A unique identifier of the page. The identifier is used by other objects to refer to a particular page.

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

name

A name of the page. The name makes it easier for users to get oriented in Electra presets.

- mandatory
- string
- minLength = 0
- maxLength = 20

defaultControlSetId

An identifier of the control set that will be selected when the page is opened.

- optional
- numeric
- default = 1
- min = 1
- max = 3

Device

A device is a MIDI hardware or software instrument connected to one of Electra's ports. It can be a hardware synth connected to a MIDI IO port, a hardware sequencer connected to Electra's USB host port, or a software plugin attached to Electra's USB device ports. Electra can handle up to 16 simultaneously connected devices. When working with Electra, you always need to define your connected devices, you never send or receive MIDI messages from port and channel directly.

example:

```
{
  "id": 1,
  "name": "Generic MIDI",
  "instrumentId": "generic-MIDI",
  "port": 1,
  "channel": 1
}
```

id

A unique identifier of the device. The identifier is used in other objects to refer to a particular device.

- mandatory
- numeric

- min = 1
- max = 16

name

A user-defined name of the device. The name makes it easier for users to remember and identify devices.

- mandatory
- string
- minLength = 0
- maxLength = 20

port

A port number that represents the MIDI bus inside the Electra. Port 1 interconnects MIDI IO port 1, USB Host port 1, USB device port 1. Port 2 interconnects MIDI IO 2, USB Host port 2, and USB device port 2.

- mandatory
- numeric
- min = 1
- max = 2

channel

A MIDI channel where the device transmits the MIDI messages.

- mandatory
- numeric
- min = 1
- max = 16

rate

A minimum wait time between sending MIDI messages to a device, specified in milliseconds.

- optional
- numeric
- min = 10
- max = 1000

Overlay

Overlays are lists of text labels linked to particular MIDI values. Typically they are assigned to selection list controls or faders. Overlays are referred to by its identifier. Overlay items can be both text labels or graphical symbols represented with bitmap images.

examples:

an overlay with text labels only


```
{
  "id":1,
  "items":[
    {
      "value": 0,
      "label": "SAW"
    },
    {
      "value": 1,
      "label": "SQUARE"
    },
    {
      "value": 2,
      "label": "WHITE NOISE"
    },
    {
      "value": 3,
      "label": "PINK NOISE"
    }
  ]
}
```

an overlay with the bitmap data

[illegible]

id

A unique identifier of the overlay. List and fader controls use the identifier to refer to a particular overlay.

- mandatory
- numeric

- min = 1
- max = 51

items

An array of value - label pairs.

- mandatory
- array

value

A MIDI value assigned to the label.

- mandatory
- numeric
- min = 0
- max = 16383

label

A text label assigned to the MIDI value. The control displays the text labels to the users.

- mandatory
- string
- minlength = 0
- maxLength = 20

bitmap

A base64 encoded bitmap image. The bitmap image is in XBM format with 48 x 18 size.

- mandatory
- base64 encoded XBM image

Group

Graphical separators used to organize controls to groups by meaning. For example a group "Envelope 1" can be created for controls "Attack", "Decay", "Sustain", and "Release". Groups do not provide any other functionality than the visual grouping of controls.

example:

```
{
  "pageId": 1,
  "name": "ENVELOPE",
  "bounds": [
    0,
    16,
    486,
```

```

    16
  ],
  "color": "FFFFFF"
}
```

id

A unique identifier of the group. Electra uses the id to uniquely identify each group. The identifiers are shared with controls.

- mandatory
- numeric
- min = 1
- max = 1023

pageId

A reference to a page identifier. Each group must belong to exactly one page and the page must be defined within the pages array.

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

name

A name of the group. The name is shown to the user inside the group graphics. The name is trimmed to fit the size of the group.

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

bounds

A bounding box of the group, ie. the definition of the group's position on the screen and its size. The bounding box is represented as an array of [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"`

Control

A Control is a representation of a MIDI parameter or a MIDI message. Controls visualize and change values of MIDI parameters. A control is for example a fader, knob, pad, or ADSR envelope. A control consists of information about values that are mapped to particular MIDI messages.

examples:

A simple control with one value assigned.

```
{
  "id": 1,
  "type": "fader",
  "name": "WHITE",
  "color": "FFFFFF",
  "variant": "thin",
  "mode": "unipolar"
  "bounds": [
    0,
    40,
    146,
    56
  ],
  "pageId": 1,
  "controlSetId": 1,
  "visible": true,
  "inputs": [
    {
      "potId": 1,
      "valueId": "value"
    }
  ],
  "values": [
    {
      "id": "value",
      "message": {
        "deviceId": 1,
        "type": "cc7",
        "parameterNumber": 1,
```

```

        "min": 0,
        "max": 127
    },
    "min": 0,
    "max": 127
}
]
}

```

an ADSR control with multiple values assigned

```

{
  "id": 1,
  "pageId": 1,
  "bounds": [
    10,
    40,
    158,
    73
  ],
  "controlSetId": 1,
  "inputs": [
    {
      "potId": 1,
      "valueId": "attack"
    },
    {
      "potId": 2,
      "valueId": "decay"
    }
  ],
  "type": "adsr",
  "name": "ADSR",
  "color": "F49500",
  "values": [
    {
      "id": "attack",
      "min": 0,
      "max": 127,
      "message": {
        "deviceId": 1,
        "type": "cc7",
        "parameterNumber": 1,
        "min": 0,
        "max": 127
      }
    },
    {
      "id": "decay",
      "min": 0,
      "max": 127,
      "message": {
        "deviceId": 1,
        "type": "cc7",

```

```

        "parameterNumber": 2,
        "min": 0,
        "max": 127
    }
},
{
    "id": "sustain",
    "min": 0,
    "max": 127,
    "message": {
        "deviceId": 1,
        "type": "cc7",
        "parameterNumber": 3,
        "min": 0,
        "max": 127
    }
},
{
    "id": "release",
    "min": 0,
    "max": 127,
    "message": {
        "deviceId": 1,
        "type": "cc7",
        "parameterNumber": 4,
        "min": 0,
        "max": 127
    }
}
]
}

```

id

A unique identifier of the control. Electra uses the id to uniquely identify each control.

- mandatory
- numeric
- min = 1
- max = 1023

type

A type of functional and visual representation of the control.

- mandatory
- enum
 - fader
 - list
 - pad
 - vfader (vertical fader)

- adsr
- adr
- dx7envelope

name

A name of the control. The name is usually shown underneath the control. When the control receives touch-event via the physical knob, the name is highlighted. If a name is an empty string or the attribute is omitted, the name is not shown and touch indication is disabled.

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

color

A 24-bit RGB code of the control's color. Electra uses 16-bit RGB565 colors internally. The conversion might affect the final displayed color.

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

variant

A visual variant of the control. Variants provide more subtle control of the visual appearance.

- optional
- string
- default = default
- enum
 - default
 - thin
 - outline
 - valueOnly
 - dial

mode

A predefined behaviour of the control. Mode does not affect how the control looks like, instead it affects how it works.

- optional
- string

- default = default
- enum
 - default
 - unipolar
 - bipolar
 - momentary
 - toggle

bounds

A bounding box of the control, ie. the definition of the control's position on the screen and its size. The bounding box is represented as an array of [x, y, width, height]

- mandatory
- array with fixed items

pageId

A reference to a page identifier. Each control must belong to exactly one page and the page must be defined within the pages array.

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

controlSetId

Controls placed on one page can be further divided into control sets. The control sets are used to assign controls to pots (knobs). Users may switch between controls sets by pressing the hardware buttons, using screen touch, or by sending MIDI messages to Electra. Only one control set can be active at any time. The controls of the active control sets are highlighted with the indicators on the sides of the screen.

- optional
- numeric
- default = 1
- min = 1
- max 3

visible

A boolean flag to manage the visibility of a Control. When set to `false`, the control is invisible.

- optional
- boolean
- default = true

values

An array of values associated with the control. A values represent an instance of the value of certain MIDI parameter. Actions made with the control (turning assigned pot, touch events) effectively change associated values and trigger transmission of MIDI messages.

- mandatory
- array

examples:

```
"values": [
  {
    "id": "value",
    "min": -64,
    "max": 63,
    "defaultValue": 0,
    "message": {
      "deviceId": 1,
      "type": "cc7",
      "parameterNumber": 1,
      "min": 0,
      "max": 127
    }
  }
]
```

Input

An Input provides information about an assignment of a physical control/gesture to a value. An example is assigning a knob to a value of the control.

- optional
- object

examples:

```
{
  "potId": 1,
  "valueId": "attack"
}
```

potId

An identifier of the physical pot (knob). There are 12 pots on Electra, identified as 1 (top-left) to 12 (bottom-right) pot. A control with an assigned pot can be controlled by turning the physical knob. Providing a given control set is active.

- mandatory
- numeric
- default = 1

- min = 1
- max = 12

valueId

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

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

Value

A value represents a parameter value of given control. A value is mapped to a value of a MIDI parameter or a MIDI message. The value object allows translation of MIDI values to user-friendly display values.

examples:

a continuous value

```
{
  "id": "value",
  "min": -64,
  "max": 63,
  "defaultValue": 0,
  "formatter": "formatFractions",
  "function": "disableOscillator",
  "message": {
    "deviceId": 1,
    "type": "cc7",
    "parameterNumber": 1,
    "min": 0,
    "max": 127
  }
}
```

value with a list of discrete values (an overlay)

```
{
  "id": "value",
  "overlayId": 2,
  "function": "displaySections",
  "message": {
    "deviceId": 1,
    "type": "cc7",
    "parameterNumber": 2
  }
}
```

```

    }
  }

```

id

An identifier of the value. This identifier is a text string. This is to make it easier for programmers to get oriented. The identifier expresses the meaning of the value, eg. attack, rate, or value.

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

min

A minimum value that the control can display. Note this is not the MIDI value, it is the minimum value that can be displayed by the control.

- optional
- numeric
- default = 0
- min = -16383
- max = 16383

max

A maximum value that the control can display. Note this is not the MIDI value, it is the maximum value that can be displayed by the control.

- optional
- numeric
- default = 0
- min = -16383
- max = 16383

defaultValue

A value to be set when the preset is loaded. The default value is also recalled when user double-taps the control on the touch screen.

- optional
- numeric
- default = 0
- min = -16383
- max = 16383

formatter

A hook to a custom Lua extension function that will be called to format the value. More information on Electra One preset Lua extension is available at [Preset Lua extension](#) document. It is up to the user what the name of the function will be, as long as the function exists in the Lua extension module. The formatting function must take exactly one parameter in (value) and must return one parameter back (formatted string).

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

function

A hook to a custom Lua extension function that will be called whenever the display value is changed. More information on Electra One preset Lua extension is available at [Preset Lua extension](#) document. It is up to the user what the name of the function will be, as long as the function exists in the Lua extension module. The function must take exactly two parameters in (controlId and value).

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

overlayId

A reference to the overlay identifier defined in the array of overlays. The list control will use the overlay items as the list items. Fader control will show overlay labels for matching values.

- optional
- numeric
- min = 1
- max = 128

message

An object that describes a MIDI message assigned to the value.

- mandatory
- object

Message

An object that describes a MIDI message that will be sent when the value of the control is changed by the touch or turning the knobs. The message is also used to parse incoming MIDI messages. When incoming MIDI data matches the message object, the value of the control is adjusted accordingly.

- optional

- object

examples:

A simple CC7 message

```
"message": {
  "deviceId": 1,
  "type": "cc7",
  "parameterNumber": 1,
  "min": 0,
  "max": 127
}
```

A message with a SysEx template

```
"message": {
  "deviceId": 1,
  "type": "SysEx",
  "parameterNumber": 6,
  "min": 0,
  "max": 127,
  "data": [
    67,
    16,
    1,
    15,
    {
      "type": "value",
      "rules": [
        {
          "parameterNumber": 40,
          "bitPosition": 0,
          "bitWidth": 3
        }
      ]
    }
  ]
}
```

A simple CC7 message with event midi values assigned

```
"message": {
  "deviceId": 1,
  "type": "cc7",
  "parameterNumber": 1,
  "offValue": 0,
  "onValue": 127
}
```

A simple NRPN message handling a negative values

```
"message": {  
  "deviceId": 1,  
  "type": "nrpn",  
  "parameterNumber": 1,  
  "lsbFirst": false,  
  "twosComplement": true,  
  "bitWidth": 7  
}
```

deviceId

A reference to the device identifier defined in the array of devices. The message will be sent to the referenced device. Also, messages received from this device that match the message definition will modify the value accordingly.

type

A type of the MIDI message. The type is not limited to basic MIDI messages but supports their higher level implementation, such as NRPN, etc.

- mandatory
- enum
 - cc7
 - cc14
 - nrpn
 - rpn
 - SysEx
 - note
 - program
 - start
 - stop
 - tune
 - atpoly
 - atchannel
 - pitchbend
 - spp

parameterNumber

A parameter number of the message. The parameterNumber is used to specify the parameter number, note number, program number, pressure, pitchbend value, song position. To fully support NRPN and SysEx, a parameterNumber is a 14-bit number.

- optional
- numeric

- min = 0
- max = 16383

min

A MIDI minimum value to be transferred. This minimum MIDI value is mapped to the display value minimum defined in the value object.

- optional
- numeric
- default = 0
- min = 0
- max = 16383

max

A MIDI maximum value to be transferred. This maximum MIDI value is mapped to the display value maximum defined in the value object.

- optional
- numeric
- default = 0
- min = 0
- max = 16383

data

An array of bytes and placeholder variables to be sent and parsed for SysEx messages.

- optional
- array

onValue

A MIDI value to be transferred when the parent control goes to active state. On a receiving side, it is the value that switches the parent control to the active state. For example, a Pad is highlighted. When onValue is not defined, MIDI transmission is ignored.

- optional
- numeric
- default = undefined (ignore)
- min = 0
- max = 16383

offValue

A MIDI value to be transferred when the parent control goes to inactive state. When offValue is not defined, MIDI transmission is ignored.

- optional
- numeric
- default = undefined (ignore)
- min = 0
- max = 16383

lsbFirst

The lsbFirst is a flag that forces Electra to swap LSB and MSB value bytes of 14-bit MIDI parameters. It can be used in combination with cc14 and nrpn MIDI messages.

- optional
- boolean
- default = false

signMode

The sigMode sets what type of negative number representation will be used for negative values.

- optional
- enum
 - none
 - twosComplement
 - signBit

bitWidth

The bitWidth specifies number of bits used to calculate the value. It tells the controller where the sign-bit should be placed for negative values.

- optional
- numeric
- default = 7 or 14 depending on the message type
- min = 1
- max = 14