

Lorby “Axis And Ohs” Gauges (Desktop and Web FIPs)

Version 3.01 - 12.04.2023

© 2018 Lorby-SI <http://www.lorby-si.com>

1 Introduction

2 Gauges

2.1 File system

2.2 Image format

2.3 Reverse Polish Notation (RPN)

2.4 Gauge Format

2.5 Variables and Event-IDs

2.6 Cursor Types for DesktopGauges

2.7 Example: Altimeter gauge

2.8 Special notes

2.9 Gauge XML Schema

1. Introduction

This document provides guidelines about how to make your own gauges for the Desktop and Web flight instruments (DesktopFIPs, WebFIPs) for the AxisAndOhs app.

Examples for gauges can be found on the AxisAndOhs website's **DOWNLOADS** page.

[Downloads - AXIS AND OHS \(weebly.com\)](#)

2. Gauges

2.1 File system

Lorby AxisAndOhs will automatically look for gauge XML files in

\Documents\AxisAndOhs Files\UserGauges

and all subfolders

\Documents\AxisAndOhs Files\UserGauges\MyGauge

\Documents\AxisAndOhs Files\UserGauges\AnotherGauge

etc.

Images that are used by the gauge must be saved to a folder called “1024” next to your gauges XML file

\Documents\AxisAndOhs Files\UserGauges\1024

\Documents\AxisAndOhs Files\UserGauges\MyGauge\1024

\Documents\AxisAndOhs Files\UserGauges\AnotherGauge\1024

etc.

2.2 Image format

- Image assets must be either in BMP or PNG 32bit color format.
- With BMP all pixels that are pure black (0,0,0) will render as transparent
- PNG files can use normal transparency, black will display as black
- Background images are usually squares with either 240x240 or 512x512 pixels. But they don't have to be, you can use custom sizes too.

2.3 Reverse Polish Notation (RPN)

RPN is a concept of the early computer days. It handles arithmetic operations not in their natural order, by based on a “stack” functionality, written in what is called “post script notation”. RPN will be necessary in the “Value” and “Visible” tags, so you can adjust the required outcome.

Please check out the chapter “Scripting” in the AxisAndOhs manual about the RPN dialect that is being used by AxisAndOhs. It is similar to the MS flightsim gauge format used in FS9, but not identical.

2.3 Gauge format

The gauges in AxisAndOhs are similar to the legacy XML gauges in FSX/P3D/MSFS. They are stored in plain XML files. Please check the SDK documentation of either simulator for specifics about the XML format – they are broadly similar and offer the same information.

The actual elements of the AAO gauges are a little bit simplified compared to proper flightsim XML gauges.

You can use the following assets:

- Gauge background picture
- Gauge Elements containing
 - Images with
 - Name
as the name of a PNG image file in subfolder "1024)
or
a Base64 String LVar ("(L:mypic, String)", created with the RPN Command "(LOADIMAGE:")
 - Axis (offset point of the image that it shall rotate around)
 - Rotation with Value and Nonlinearity (Image should rotate by Value, possibly in a non-linear way)
 - Shift with Value and Nonlinearity (Image should be shifted by Value, possibly in a non-linear way)
 - Mask only part of the image should be displayed

- Gauge text, can be combined with Rotation and Shift
- Visibility script as `<Visible>RPN code</Visible>`
- WebFIPs can also contain a `<Redirect>` element. This will create an iFrame in the gauge using the size of the background image and then call the URL supplied in the element (hard coded or as an RPN script that returns a string value). This can be used for example with the InGame Panel in MSFS to display other web-based assets inside the panel at a certain location and with a specific size.
- Mouse areas containing
 - Position and Size: if those are omitted, the app will assign the mouse areas to the lower half of the gauge
 - Click and a separate Release (=Mouse Down, Mouse Up)
 - Wheel
 - Cursor shape
 - Visibility script as `<Visible>RPN code</Visible>`

2.4 Variables and Event- IDs

Lorby AxisAndOhs gauges can utilize the variables and event ids that are listed in the SDK documentation of the simulators.

Currently supported are A:, L: and E: variables, plus K: events. You can also trigger an AxisAndOhs script as a K: event, as described in the AAO manual. The custom AAO “S:” notation is also available, to evaluate RPN code directly.

2.5 Cursor Types

By adding a “<Cursor Type=...”/> tag to a Mouse Area, that cursor will be shown when you hover the mouse over it. You can set the following types. They may appear differently in Desktop and Web FIPs.

| Type | Description (Desktop FIP) | Shown on a WebFIP as CSS |
|-------------|---|--------------------------|
| AppStarting | Gets the Cursor that appears when an application is starting. | cursor pointer |
| Arrow | Gets the Arrow Cursor. | default |
| Cross | Gets the crosshair Cursor. | crosshair |
| Hand | Gets a hand Cursor. | pointer |
| Help | Gets a help Cursor which is a combination of an arrow and a question mark. | help |
| IBeam | Gets an I-beam Cursor, which is used to show where the text cursor appears when the mouse is clicked. | text |
| No | Gets a Cursor with which indicates that a particular region is invalid for a given operation. | not-allowed |
| None | Gets a special cursor that is invisible. | none |
| Pen | Gets a pen Cursor. | pointer |
| ScrollAll | Gets the scroll all Cursor. | all-scroll |
| ScrollE | Gets the scroll east Cursor. | e-resize |
| ScrollN | Gets the scroll north Cursor. | n-resize |
| ScrollNE | Gets the scroll northeast cursor. | ne-resize |
| ScrollNS | Gets the scroll north/south cursor. | ns-resize |
| ScrollNW | Gets a scroll northwest cursor. | nw-resize |
| ScrollS | Gets the scroll south Cursor. | s-resize |
| ScrollSE | Gets a south/east scrolling Cursor. | se-resize |
| ScrollSW | Gets the scroll southwest Cursor. | sw-resize |
| ScrollW | Gets the scroll west Cursor. | w-resize |

| | | |
|----------|---|-------------|
| ScrollWE | Gets a west/east scrolling Cursor. | ew-resize |
| SizeAll | Gets a four-headed sizing Cursor, which consists of four joined arrows that point north, south, east, and west. | all-scroll |
| SizeNESW | Gets a two-headed northeast/southwest sizing Cursor. | nesw-resize |
| SizeNS | Gets a two-headed north/south sizing Cursor. | ns-resize |
| SizeNWSE | Gets a two-headed northwest/southeast sizing Cursor. | nwse-resize |
| SizeWE | Gets a two-headed west/east sizing Cursor. | ew-resize |
| UpArrow | Gets an up arrow Cursor, which is typically used to identify an insertion point. | row-resize |
| Wait | Specifies a wait (or hourglass) Cursor. | wait |

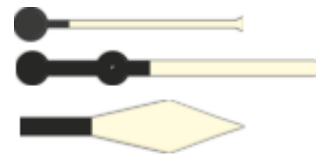
2.6 Example

Since gauges are somewhat daunting at first, here is a simple example: the altimeter. This instrument has the following assets:

- a fixed background picture “AltimeterBackground.png”
240x240 pixel



- three pictures of needles (100, 1000 and 10000ft) which will rotate around their axis points by the appropriate fraction of the current altitude (A:Indicated Altitude, Feet)



- a picture of the QNH knob “AltDial.png” which will rotate around its center in sync with the current QNH value (A:Kohlsman Setting mb, Millibars)



- a GaugeText Element to display the QNH setting in the little black rectangle of the background picture

- Three mouse areas for click and mousewheel events:
Click on the left half of the knob picture to dial down the QNH, the right half to increase it, or spin the mousewheel anywhere on the knob.

```
<?xml version="1.0" encoding="utf-8"?>
<Gauge Name="Altimeter MB" Version="1.0">
<Image Name="AltimeterBackground.png" ImageSizes="240,240,240,240"/>
<Element>
  <Position X="163" Y="120"/>
  <GaugeText>
    <FontFace>COURIER NEW</FontFace>
    <FontHeight>16</FontHeight>
    <FontColor>0xffffffff</FontColor>
    <Bold>>true</Bold>
    <Value>%(A:Kohlsman setting mb, Millibars)!0.0f!</Value>
  </GaugeText>
</Element>
<Element>
  <Position X="120" Y="120"/>
  <Image Name="AltimeterNeedleTenThousand.png" ImageSizes="114,26,114,26">
    <Axis X="8" Y="13" PointsTo="North"/>
  </Image>
  <Rotate>
    <Value>(A:Indicated Altitude, feet) 10000 </Value>
    <Nonlinearity>
      <Item Value="0" Degrees="0" />
      <Item Value="5" Degrees="180" />
    </Nonlinearity>
  </Rotate>
</Element>
<Element>
  <Position X="120" Y="120"/>
```

```

<Image Name="AltimeterNeedleThousand.png" ImageSizes="112,26,112,26">
  <Axis X="15" Y="12" PointsTo="North"/>
</Image>
<Rotate>
  <Value>(A:Indicated Altitude, feet) 1000 </Value>
  <Nonlinearity>
    <Item Value="0" Degrees="0" />
    <Item Value="5" Degrees="180" />
  </Nonlinearity>
</Rotate>
</Element>
<Element>
  <Position X="120" Y="120"/>
  <Image Name="AltimeterNeedleHundred.png" ImageSizes="157,19,157,19">
    <Axis X="51" Y="9.5" PointsTo="North"/>
  </Image>
  <Rotate>
    <Value>(A:Indicated Altitude, feet) 100 </Value>
    <Nonlinearity>
      <Item Value="0" Degrees="0" />
      <Item Value="5" Degrees="180" />
    </Nonlinearity>
  </Rotate>
</Element>
<Element>
  <Position X="215" Y="215" />
  <Image Name="AltDial.png" ImageSizes="48,48,48,48">
    <Axis X="24" Y="24"/>
  </Image>
  <Rotate>
    <Value Minimum="28.1" Maximum="31.5">(A:Kohlsman setting hg,inHg)</Value>
    <Nonlinearity>
      <Item Value="28" Degrees="720" />
      <Item Value="31" Degrees="0" />
    </Nonlinearity>
  </Rotate>
</Element>

```

```
        </Rotate>
</Element>
<Mouse>
  <Area>
    <Area>
      <Position X="190" Y="190"/>
      <Size X="50" Y="50"/>
      <Wheel DownEvent="KOHLSMAN_DEC" UpEvent="KOHLSMAN_INC"/>
      <Cursor Type="ScrollWE"/>
    </Area>
    <Area>
      <Position X="190" Y="190"/>
      <Size X="25" Y="50"/>
      <Click Event="KOHLSMAN_DEC" />
    </Area>
    <Area>
      <Position X="215" Y="190"/>
      <Size X="25" Y="50"/>
      <Click Event="KOHLSMAN_INC"/>
    </Area>
  </Area>
</Mouse>
</Gauge>
```

2.7 Special notes

- the `<ImageSizes>` tags do not scale the image. They should always contain the actual size of the picture.
- `<NonLinearity>` keys and values must be numerically ascending (so “Degrees=” values can/must be greater than 360 if necessary)
- If `<Position>` and `<Size>` are omitted for mouse areas, the app will distribute them evenly over the lower half of the gauge automatically.
- Color codes for GaugeText are in HTML hexadecimal RRGGBB format, for example 0xff0000 is the code for pure red. There are a few web sites where HTML color codes can be found and calculated. When a script is used to provide a color code (as in `<FontColorScript>`), the result must be numeric (decimal or hex):

```
<FontColorScript>(L:tes) 1 == if{ 0xff0000 } els{ 0x00ff00 }</FontColorScript>
```

Transparency/alpha is not supported.

- `<Input>` element considerations:
 - the `<Variable>` tag the variable name without parentheses:
`<Variable>L:mylvar</Variable>` for numerical input
`<Variable>L:mylstringvar, String</Variable>` for alphanumerical input
 - the `<Value>` tag can only be used to set the initial value of the input element.
 - All input is written to the variable upon each key press.

2.8 Deploy a gauge

Best strategy is to create a subfolder of your own in Documents\LorbyAxisAndOhs Files\UserGauges, with a subfolder \1024 for the graphics:

\UserGauges\MyGauge\mygauge.xml

(...)

\UserGauges\MyGauge\1024\mypic1.png

\UserGauges\MyGauge\1024\mypic2.png

(...)

AAO will find new gauges only on startup, so if you want to deploy a gauge that AAO is unfamiliar with, shut down and restart the app.

When you are editing an existing gauge XML, you can reload it either by disconnecting/reconnecting AAO to the sim (click on the green LED at the top of the main AAO window) or using the “Reload” button on the DesktopFIP Handler.

WebFIPs and DesktopFIPs have the same format. So it is advisable to develop WebFIPs on the Desktop first. You can build your entire web layout on the desktop and then import it into the web part when it is finished using the “Layout” button.

2.9 XML Schema / xsd

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Gauge" nillable="true" type="GaugeDefinition" />
  <xs:complexType name="GaugeDefinition">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="Visible" type="Visible" />
      <xs:element minOccurs="0" maxOccurs="1" name="Redirect" type="Redirect" />
      <xs:element minOccurs="0" maxOccurs="unbounded" name="Update" type="Update" />
      <xs:element minOccurs="0" maxOccurs="unbounded" name="Macro" type="Macro" />
      <xs:element minOccurs="0" maxOccurs="1" name="Image" type="Image" />
      <xs:element minOccurs="0" maxOccurs="unbounded" name="Element" type="Element" />
      <xs:element minOccurs="0" maxOccurs="unbounded" name="Mouse" type="Mouse" />
    </xs:sequence>
    <xs:attribute name="Name" type="xs:string" />
    <xs:attribute name="Version" type="xs:string" />
  </xs:complexType>
  <xs:complexType name="Visible">
    <xs:simpleContent>
      <xs:extension base="xs:string" />
    </xs:simpleContent>
  </xs:complexType>
  <xs:complexType name="Redirect">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="Url" type="xs:string" />
      <xs:element minOccurs="0" maxOccurs="1" name="UrlScript" type="Value" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="Value">
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="Minimum" type="xs:string" />
        <xs:attribute name="Maximum" type="xs:string" />
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
  <xs:complexType name="Update">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="Script" type="xs:string" />
      <xs:element minOccurs="1" maxOccurs="1" name="Frequency" type="xs:int" />
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

```

</xs:sequence>
</xs:complexType>
<xs:complexType name="Macro">
  <xs:simpleContent>
    <xs:extension base="xs:string">
      <xs:attribute name="Name" type="xs:string" />
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
<xs:complexType name="Image">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="Axis" type="Axis" />
    <xs:element minOccurs="0" maxOccurs="1" name="Nonlinearity" type="Nonlinearity" />
  </xs:sequence>
  <xs:attribute name="Name" type="xs:string" />
  <xs:attribute name="PointsTo" type="xs:string" />
  <xs:attribute name="ImageSizes" type="xs:string" />
</xs:complexType>
<xs:complexType name="Axis">
  <xs:attribute name="X" type="xs:float" use="required" />
  <xs:attribute name="Y" type="xs:float" use="required" />
  <xs:attribute name="PointsTo" type="xs:string" />
</xs:complexType>
<xs:complexType name="Nonlinearity">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="Item" type="NonLinearityItem" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="NonLinearityItem">
  <xs:attribute name="Value" type="xs:double" use="required" />
  <xs:attribute name="X" type="xs:double" use="required" />
  <xs:attribute name="Y" type="xs:double" use="required" />
  <xs:attribute name="degrees" type="xs:string" />
  <xs:attribute name="Degrees" type="xs:string" />
</xs:complexType>
<xs:complexType name="Element">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="Position" type="Position" />
    <xs:element minOccurs="0" maxOccurs="1" name="MaskImage" type="Image" />
    <xs:element minOccurs="0" maxOccurs="1" name="Image" type="Image" />
    <xs:element minOccurs="0" maxOccurs="1" name="Shift" type="Shift" />
    <xs:element minOccurs="0" maxOccurs="1" name="ShiftB" type="Shift" />
    <xs:element minOccurs="0" maxOccurs="1" name="Rotate" type="Rotate" />
  </xs:sequence>

```

```

    <xs:element minOccurs="0" maxOccurs="1" name="Roll" type="Roll" />
    <xs:element minOccurs="0" maxOccurs="1" name="GaugeText" type="GaugeText" />
    <xs:element minOccurs="0" maxOccurs="1" name="Input" type="Input" />
    <xs:element minOccurs="0" maxOccurs="1" name="Visible" type="Visible" />
    <xs:element minOccurs="0" maxOccurs="1" name="SizeScale" type="xs:string" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="Position">
  <xs:attribute name="X" type="xs:float" use="required" />
  <xs:attribute name="Y" type="xs:float" use="required" />
</xs:complexType>
<xs:complexType name="Shift">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="Value" type="Value" />
    <xs:element minOccurs="0" maxOccurs="1" name="Scale" type="Scale" />
    <xs:element minOccurs="0" maxOccurs="1" name="Nonlinearity" type="Nonlinearity" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="Scale">
  <xs:attribute name="X" type="xs:float" use="required" />
  <xs:attribute name="Y" type="xs:float" use="required" />
</xs:complexType>
<xs:complexType name="Rotate">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="Value" type="Value" />
    <xs:element minOccurs="0" maxOccurs="1" name="Nonlinearity" type="Nonlinearity" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="Roll">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="Value" type="Value" />
    <xs:element minOccurs="0" maxOccurs="1" name="Scale" type="Scale" />
    <xs:element minOccurs="0" maxOccurs="1" name="Nonlinearity" type="Nonlinearity" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="GaugeText">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="FontFace" type="xs:string" />
    <xs:element minOccurs="1" maxOccurs="1" name="FontHeight" type="xs:int" />
    <xs:element minOccurs="0" maxOccurs="1" name="FontHeightScript" type="Value" />
    <xs:element minOccurs="0" maxOccurs="1" name="FontColor" type="xs:string" />
    <xs:element minOccurs="0" maxOccurs="1" name="FontColorScript" type="Value" />
    <xs:element minOccurs="1" maxOccurs="1" name="Bold" type="xs:boolean" />
  </xs:sequence>
</xs:complexType>

```

```

    <xs:element minOccurs="0" maxOccurs="1" name="HorizontalAlign" type="xs:string" />
    <xs:element minOccurs="0" maxOccurs="1" name="Value" type="Value" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="Input">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="FontFace" type="xs:string" />
    <xs:element minOccurs="1" maxOccurs="1" name="FontHeight" type="xs:int" />
    <xs:element minOccurs="1" maxOccurs="1" name="Width" type="xs:int" />
    <xs:element minOccurs="0" maxOccurs="1" name="FontHeightScript" type="Value" />
    <xs:element minOccurs="0" maxOccurs="1" name="FontColor" type="xs:string" />
    <xs:element minOccurs="0" maxOccurs="1" name="FontColorScript" type="Value" />
    <xs:element minOccurs="1" maxOccurs="1" name="Bold" type="xs:boolean" />
    <xs:element minOccurs="0" maxOccurs="1" name="HorizontalAlign" type="xs:string" />
    <xs:element minOccurs="0" maxOccurs="1" name="Value" type="Value" />
    <xs:element minOccurs="0" maxOccurs="1" name="Variable" type="xs:string" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="Mouse">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="Area" type="Area" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="Area">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="Area" type="Area" />
    <xs:element minOccurs="0" maxOccurs="1" name="Visible" type="Visible" />
    <xs:element minOccurs="0" maxOccurs="1" name="Position" type="Position" />
    <xs:element minOccurs="0" maxOccurs="1" name="Size" type="Size" />
    <xs:element minOccurs="0" maxOccurs="1" name="Cursor" type="Cursor" />
    <xs:element minOccurs="0" maxOccurs="1" name="Click" type="Click" />
    <xs:element minOccurs="0" maxOccurs="1" name="Release" type="Release" />
    <xs:element minOccurs="0" maxOccurs="1" name="Wheel" type="Wheel" />
    <xs:element minOccurs="1" maxOccurs="1" name="Repeat" type="xs:boolean" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="Size">
  <xs:attribute name="X" type="xs:int" use="required" />
  <xs:attribute name="Y" type="xs:int" use="required" />
</xs:complexType>
<xs:complexType name="Cursor">
  <xs:attribute name="Type" type="xs:string" />
</xs:complexType>

```

```
<xs:complexType name="Click">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="Value" type="Value" />
  </xs:sequence>
  <xs:attribute name="Event" type="xs:string" />
  <xs:attribute name="Repeat" type="xs:string" />
</xs:complexType>
<xs:complexType name="Release">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="Value" type="Value" />
  </xs:sequence>
  <xs:attribute name="Event" type="xs:string" />
</xs:complexType>
<xs:complexType name="Wheel">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="ValueUp" type="Value" />
    <xs:element minOccurs="0" maxOccurs="1" name="ValueDown" type="Value" />
  </xs:sequence>
  <xs:attribute name="UpEvent" type="xs:string" />
  <xs:attribute name="DownEvent" type="xs:string" />
</xs:complexType>
</xs:schema>
```