

LightningChart JS Migration Guide

From version 2.2.1 to version 3.0.0

ChartXY.addAxisX/Y

The parameters of addAxis methods have changed. If you used to supply **no arguments**, usage has not changed.

Before 3.0 there was a single `addTop` Boolean parameter. Moving to 3.0, migrate it like follows:

Usage before 3.0:

```
Chart.addAxisX(true)
```

Usage after 3.0:

```
Chart.addAxisX({ opposite: true })
```

Scale changes

UI Element custom scale changes

Concerns following methods:

- `Chart.addUIElement`
- `Chart.addLegendBox`
- `Dashboard.addUIElement`
- `Dashboard.addLegendBox`

Changes apply only to the second parameter, **scale**. This parameter was and still is optional – if you did not use it, or supplied *undefined*, no changes are necessary.

Positioning UI elements on Axis values

Previous versions syntax:

```
chart.addUIElement(UIElementBuilders.TextBox, { x: xAxis.scale, y: yAxis.scale })
```

LCJS v3.0 syntax:

```
chart.addUIElement(UIElementBuilders.TextBox, { x: xAxis, y: yAxis })
```

Positioning UI elements on different X and Y scales

Previous versions syntax:

```
chart.addUIElement(UIElementBuilders.TextBox, { x: chart.uiScale.x, y: chart.pixelScale.y })
```

LCJS v3.0 syntax:

```
chart.addUIElement(UIElementBuilders.TextBox, { x: chart.uiScale, y: chart.pixelScale })
```

Removed Axis properties

Property	Migration instructions
getTickStyle()	Default Axis tick style can be looked up via <code>Themes.dark.numericTickStrategy</code>
plottingScale	Use Axis itself instead, see "Positioning UI elements on Axis values"
heightScale	Use Axis itself instead, see "Positioning UI elements on Axis values"

Axis.scale

`Axis.scale` property has been removed.

Property	Migration instructions
<code>Axis.scale.getInnerStart()</code>	<code>Axis.getInterval().start</code>
<code>Axis.scale.getInnerEnd()</code>	<code>Axis.getInterval().end</code>
<code>Axis.scale.getInnerInterval()</code>	<code>Math.abs(Axis.getInterval().end - Axis.getInterval().start)</code>
<code>Axis.scale.getPixelSize()</code>	No direct replacement. If you copied the usage from Arction examples, refer to the example for up-to-date usage.
<code>Axis.scale.getCellSize()</code>	No direct replacement. If you copied the usage from Arction examples, refer to the example for up-to-date usage.

Translating Axis coordinates to other coordinate systems, and other way around `translatePoint` now accepts `Axis` object as is, instead of previous way of using `Axis.scale`.

Example usage for translating an axis coordinate to pixels.

```
const pixelLocation = translatePoint(
  // axis coordinate.
  { x: 116.9, y: 26.4 },
  {
    x: chart.getDefaultAxisX(),
    y: chart.getDefaultAxisY(),
  },
  chart.pixelScale
)
```

Removed Scale API

Concerns following properties:

- Chart.uiScale
- Chart.pixelScale
- Chart.engine.scale
- Dashboard.uiScale
- Dashboard.engine.scale
- Series.scale

All these properties' APIs have been removed.

Property	Migration instructions
uiScale	getInnerStart() is equal to 0. getInnerEnd() is equal to 100.
Series.scale	Use Axis.getInterval() instead.
getCellSize()	No direct replacement. If you copied the usage from Arction examples, refer to the example for up-to-date usage.
getPixelSize()	No direct replacement. If you copied the usage from Arction examples, refer to the example for up-to-date usage.

DataPattern

Syntax of DataPattern API has been changed. Now it utilizes more flexible object parameters. New option regularProgressiveStep has been added for even more application specific optimizations.

LCJS v2.2.1:

```
const lineSeries = ChartXY.addLineSeries({
  dataPattern: DataPatterns.horizontalProgressive
})
```

LCJS v3.0.0:

```
const lineSeries = ChartXY.addLineSeries({
  dataPattern: {
    // pattern: 'ProgressiveX' => Each consecutive data point has
    // increased X coordinate.
    pattern: 'ProgressiveX',
    // regularProgressiveStep: false => The X step between each
    // consecutive data point is not regular.
    regularProgressiveStep: false,
  }
})
```

In previous versions, `dataPattern` also selected basis of solving the nearest data point from mouse for cursor functionality. This functionality has been moved to a dedicated API, *cursor solve basis*, which can be set for `LineSeries`, `PointLineSeries`, `SplineSeries` and `StepSeries`:

```
const lineSeries = ChartXY.addLineSeries()  
  .setCursorSolveBasis('nearest-x')
```

Supported values are `'nearest-x'`, `'nearest-y'` and `'nearest'`

LegendBox changes

LegendBox title

Previous behavior: LegendBox title is automatically set to match title of Chart

New behavior: LegendBox title is empty by default. Set with `LegendBox.setTitle('My title')`

LegendBox.add parameter changes

disposeOnClick

Previous syntax: `LegendBox.add(component, false)`

3.0 syntax: `LegendBox.add(component, { disposeOnClick: false })`

tag

Previous syntax: `LegendBox.add(component, undefined, 'Group tag')`

3.0: No direct replacement. To add a title to *LegendBox*, use `LegendBox.setTitle('Group tag')`

builder

Previous syntax: `LegendBox.add(component, undefined, undefined, entryBuilder)`

3.0 syntax: `LegendBox.add(component, { builder: entryBuilder })`

Changes to styling LegendBoxEntries

Previous syntax:

```
const entries = LegendBox.add( chart )  
entries.forEach(entry => entry.setTextFillStyle( ... ))
```

3.0 syntax:

```
LegendBox.add( chart ).setEntries( entry => entry.setTextFillStyle( ... )  
)
```

UILegendBoxPanel

add()

Series or other *attachables* can no longer be directly added to a *legendBox panel*. Only whole charts, or dashboard containing one or several charts can be added.

title

legendBox panel no longer has a built-in title component. A custom title can be added with `UILegendBoxPanel.addUIElement`

Each attached chart has its own internally created `LegendBox`, each of which also has a configurable title.

Removed methods:

- `UILegendBoxPanel.setTitle`
- `UILegendBoxPanel.getTitle`
- `UILegendBoxPanel.setTitleFillStyle`
- `UILegendBoxPanel.getTitleFillStyle`
- `UILegendBoxPanel.setTitleFont`
- `UILegendBoxPanel.getTitleFont`

setEntries()

Legendbox entries are now grouped into separate *legendboxes*, grouped by charts. To access *legendbox entries* inside a *legendbox panel*, use `UILegendBoxPanel.setLegendBoxes`. See example usage documentation [here](#).

Custom tick changes

`CustomTick.setTopPadding` has been renamed to `setTickLabelPadding`

`PointableTextBox` has been renamed to `UIPointableTextBox`

`CustomTick.setPaddingBottom` and `CustomTick.setSidePaddings` have been **removed**. They can be accessed via *CustomTick Marker background* (if the tick marker has a background).

```
CustomTick.setMarker((tickMarker: UIPointableTextBox) => tickMarker
    .setBackground((background) => background
        .setPadding({
            // Padding bottom
            bottom: 10,
            // Side paddings
            left: 20,
            right: 20
        })
    )
)
```

General changes

- Removed `PointSeriesOptions3D.pointShape`. Use `PointSeries3D.setPointStyle` instead.
- Removed `PointLineSeriesOptions3D.pointShape`. Use `PointLineSeries3D.setPointStyle` instead.
- Removed `Theme.numericTickStrategy3D`. Use `Theme.numericTickStrategy` instead.
- Removed `Theme.dateTickStrategy3D`. Use `Theme.dateTimeTickStrategy` instead.
- `ResultTable.setFont` renamed to `setTextFont`
- `UITextBox.setFont` renamed to `setTextFont`
- `UICheckBox.setFont` renamed to `setTextFont`
- `LegendBoxEntry.setFont` renamed to `setTextFont`
- `setChartBackgroundFillStyle` methods renamed to `setSeriesBackgroundFillStyle`
- `getChartBackgroundFillStyle` methods renamed to `getSeriesBackgroundFillStyle`
- `setChartBackgroundStrokeStyle` methods renamed to `setSeriesBackgroundStrokeStyle`
- `getChartBackgroundStrokeStyle` methods renamed to `getSeriesBackgroundStrokeStyle`
- `on/offChartBackground...` event methods were renamed to `on/offSeriesBackground...`
- `setResultTableFormatter` renamed to `setCursorResultTableFormatter`
- `getResultTableFormatter` renamed to `getCursorResultTableFormatter`
- `Theme.chartBackgroundFillStyle` renamed to `Theme.seriesBackgroundFillStyle`
- `Theme.chartBackgroundStrokeStyle` renamed to `Theme.seriesBackgroundStrokeStyle`