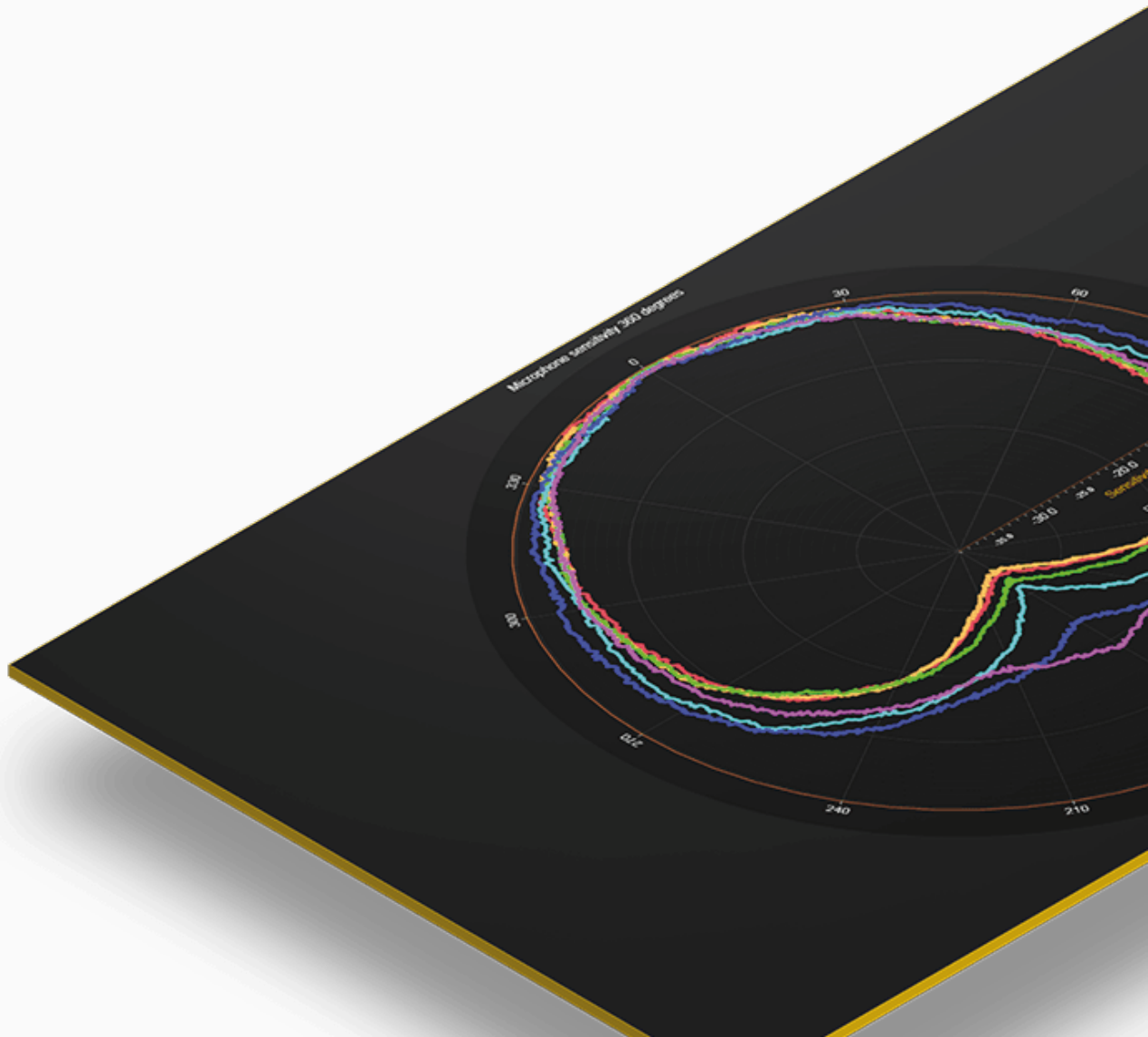


# CHANGE LOG.

---

- ➔ All notable changes to this project will be documented in this file.
- ➔ The format is based on [Keep a ChangeLog](#), and this project adheres to [Semantic Versioning](#).



 **[3.2.0] - 2021-09-22****### Added**

- Added `DateTickStrategy.setCursorFormatter`` for easy modification of cursor ticks formatting.
- Added `NumericTickStrategy.setCursorFormatter`` for easy modification of cursor ticks formatting.
- Added `TimeTickStrategy.setCursorFormatter`` for modification of cursor ticks formatting.
- Added `axis.setThickness``.
  - This can be used to control the width or height of axis depending on is the axis Y or X axis.
- Added primitive draw mode to `LineSeries3D`` and `PointLineSeries3D``.
  - This feature can be used for improved performance by reducing line depth perception.
  - See `setStrokeStyle`` method for more information.
- Added `series.setAutoScrollingEnabled`` for disabling scrolling and fitting of specific series.
- Added `synchronizeAxisIntervals``. Utility for synchronizing intervals of one or more axis.

**### Changed**

- Improved default Date time cursor formatting to display Year, month, day, hour and minute information. This can be modified with new method `DateTickStrategy.setCursorFormatter``.
- Added `dataPoint`` parameter to `SeriesXYFormatter``. This allows using `value``, `size``, `color``, `rotation`` and custom values in cursor result table formatters.
- `ZoomBandChart`` now automatically aligns itself with the reference chart.
- Community version no longer has performance penalty.
- `NumericTickStrategy.setFormattingFunction`` now also affects default cursor result table formatting and cursor ticks formatting.
- `SolidGauge.setIntervalLabelFormatter`` and `setDataLabelFormatter`` now also accept callback functions which can be used to display text that is not strictly numbers.
- Improved axis tick rendering performance.

**### Fixed**

- Fixed ChartXY UI element positioning on axis.
- Fixed map chart AutoCursor not usable with touch.
- Fixed `ZoomBandChart`` not reacting to `Dashboard`` `theme`` or `disableAnimations`` options.
- `PointSeries3D`` no longer crashes when styled with `PalettedFill``.
- `PointSeries3D`` now displays color lookup table range when attached to a legend box and styled with `paletted fill``.
- Fixed edge case in text rendering when creating text with large font size.
- Fixed occasional error message when zooming in and out line series rapidly.
- Fixed incorrect rendering behavior when chart is transitioned to fullscreen on Safari.
- Fixed incorrect legend box entry alignment after restoring the legend box entry.
- Fixed `OHLCSeries`` sometimes not being visible, especially when chart is created and animations are disabled.
- Fixed point, line and range series `getPointAmount`` not counting points that were added just before.
- Fixed empty series affecting automatic fitting of series.
- LightningChart JS logo now renders properly on high DPI devices with `window.devicePixelRatio`` greater than 1.
- Fixed poor performance when adding new points to charts with 1 million or more points in some cases.
- Fixed `HeatmapGridSeries.getWireframeStyle`` returning `FillStyle`` instead of `LineStyle``.
- Fixed TypeScript issue with being able to pass `LineStyle`` as `FillStyle`` or vice versa.
- Fixed `HeatmapScrollingGridSeries`` not attaching to any other axis than default chart axis.
- Fixed `IntensityGridSeries`` and `IntensityMeshSeries`` not rendering after it has been restored.



 **[3.1.0] - 2021-07-28****### Added**

- Added new chart type, `MapChart`.
  - `MapChart` has multiple views for visualizing data in different regions of world.
  - Supports real-time data updates with high FPS.
- Added new series type, `PolygonSeries`.
  - This series can be created with `ChartXY.addPolygonSeries`.
  - `PolygonSeries.add` returns you a `PolygonFigure` object which can be used to style the polygon.
- Added improved series type `HeatmapGridSeries`.
  - This series can be created with `ChartXY.addHeatmapGridSeries`.
  - Replaces the existing `IntensityGridSeries` (`ChartXY.addHeatmapSeries`) which is now considered deprecated and will be removed in next major release.
  - This new series type provides considerably better performance compared to the previous version.
- Added new series type `HeatmapScrollingGridSeries`.
  - This series can be created with `ChartXY.addHeatmapScrollingGridSeries`.
  - Provides a way to create scrolling heatmap with data cleaning support.
- Added new methods to help to make better experience on devices with smaller screens.
  - `UIElement.setAutoDispose`, `UIElement.getAutoDispose`
- Added `High precision XY Axis`.
  - New XY Axis type that allows zooming to smaller axis intervals with drawback of decreased performance. Refer to `AxisOptions` documentation for instructions.
  - This is considered an experimental feature and might be changed in a minor or major release with no backwards compatibility.
- Added `AxisTickStrategies.Time`.
  - New automatic ticks strategy for depicting axis intervals between hundreds of hours and individual nanoseconds.
- Added `TimeFormattingFunctions` helper methods for formatting millisecond time stamps to different time string formats.
- Added `LUT` parameter to `UILUTCheckBox.setLUTStepValueFormatter` callback.
- Added new options to `LUT`, `LUT.valueRangeMin` and `LUT.valueRangeMax`.



- Added new control for suppressing warnings that are mostly useful when developing, ``LightningChartOptions.warnings``.
- Added ``NumericTickStrategy.formattingOffset``, works similarly to ``DateTimeTickStrategy.setDateOrigin``.
- Ability to rotate all texts.
  - ``Chart.setTitleRotation``
  - ``Axis.setTitleRotation``
  - ``VisibleTicks.setLabelRotation``
  - ``ResultTable.setTextRotation``
  - ``CustomTick.setTickLabelRotation``
  - ``LegendBox.setTitleRotation``
  - ``UITextBox.setTextRotation``
- Added helper method ``Chart.zoom``.
- Added helper method ``Chart.pan``.
- Added missing API ``PolarChart.getSeries``.
- Added missing API ``Chart3D.getSeries``.
- Added ability to control mouse and touch interaction state to 3D Chart.
  - ``Chart3D.setMouseInteractions``
  - ``Chart3D.setMouseInteractionZoom``, ``Chart3D.getMouseInteractionZoom``
  - ``Chart3D.setMouseInteractionRotate``, ``Chart3D.getMouseInteractionRotate``
- Added ``LineSeries3D.setStrokeStyle, getStrokeStyle, setStrokeStyleHighlight, getStrokeStyleHighlight``.
  - These replace the similarly named ``LineSeries3D.setLineStyle, getLineStyle, setLineStyleHighlight, getLineStyleHighlight``.
- Added ``PointLineSeries3D.setStrokeStyle, getStrokeStyle, setStrokeStyleHighlight, getStrokeStyleHighlight``.
  - These replace the similarly named ``PointLineSeries3D.setLineStyle, getLineStyle, setLineStyleHighlight, getLineStyleHighlight``.
- Added ``individualPointColorEnabled``, ``individualPointSizeEnabled`` and ``individualLookupValuesEnabled`` to ``PointSeriesOptions3D``.
- Added ``UIElementBuilders.LUTRange``.
- Added more control to how 3D series are shaded.
  - ``Series3D.setColorShadingStyle``, ``Series3D.getColorShadingStyle``
- Added new Shading style ``ColorShadingStyles.Simple``.
  - This shading style can make some 3D visualizations look better by removing all shading and using the object colors directly.
  - Using this shading style can improve performance on low end devices.
- Added new Shading style ``ColorShadingStyles.Phong``.
  - This has been the default shading style for all 3D features since the 3D features were introduced.
  - You can now control ``ambientColor``, ``ambientReflection``, ``diffuseReflection``, ``specularReflection``, ``specularColor`` and ``shininess`` options.



- Added `ChartXY.addOnScreenMenu``.
  - `OnScreenMenu`` can be used to create UI controls over the chart.
  - This is considered an experimental feature and might be changed in a minor or major release with no backwards compatibility.
- Added new theme styling options.
  - `Theme.lineSeries3DStrokeStyle``
  - `Theme.pointSeries3DPointStyle``
  - `Theme.pointLineSeries3DStrokeStyle``
  - `Theme.pointLineSeries3DPointStyle``
  - `Theme.pointCloudSeries3DPointStyle``
  - `Theme.polygonSeriesFillStyle``
  - `Theme.polygonSeriesStrokeStyle``
  - `Theme.heatmapFillStyle``
  - `Theme.heatmapWireframeStyle``
  - `Theme.heatmapGridSeriesFillStyle``
  - `Theme.heatmapGridSeriesWireframeStyle``
  - `Theme.rectangleSeriesFillStyle``
  - `Theme.rectangleSeriesStrokeStyle``
  - `Theme.polarAreaSeriesFillStyle``
  - `Theme.polarAreaSeriesStrokeStyle``
  - `Theme.polarAreaSeriesStrokeStyle``
  - `Theme.uiPointableTextBoxFillStyle``
  - `Theme.uiPointableTextBoxTextFillStyle``
  - `Theme.uiPointableTextBoxStrokeStyle``
  - `Theme.uiPointableTextBoxFont``
  - `Theme.uiTickTextFillStyle``
  - `Theme.uiTickStrokeStyle``
  - `Theme.uiTickFont``
- Added new themes.
  - `Themes.darkGold``
  - `Themes.darkGreen``
  - `Themes.darkLime``
  - `Themes.darkMagenta``
  - `Themes.darkRed``
  - `Themes.darkTurquoise``
  - `Themes.blueSciFiNew``
  - `Themes.glacier``
  - `Themes.lightNew``
  - `Themes.lightNature``
  - `Themes.darkNature``
  - `Themes.duskInLapland``
  - `Themes.auroraBorealisNew``
  - `Themes.cyberSpace``
  - Themes that end with suffix  ``New`` will replace similarly named theme with no  ``New`` suffix.
- Most Theme series style properties now support Palette definition optionally as well.



### ### Changed

- `FormattingFunctions.Numeric`` now also works for more than 3 decimal parts.
- `PointSeries3D`` now supports `IndividualPointFill`` style and `PalettedFill`` style.
- `PointSeries3D`` performance has been greatly improved overall.
- `LineSeries3D`` performance has been greatly improved overall.
- `PointLineSeries3D`` performance has been greatly improved overall.
- Removed `_empty UI backgrounds_` from all default UI builders. This makes it easier to style UI backgrounds - afterwards, only using `setBackground(bg => bg.setFillStyle(...))`` will always be enough.
- Changed UI default background behavior - now ALL UI elements will always have background styled according to Theme by default. To hide background, use `setBackground(bg => bg.setFillStyle(...))``, etc.
- `DateTickStrategy`` default cursor formatting now falls back to major ticks formatting if minor ticks are disabled.
- Improved LUT precision.

### ### Removed

### ### Fixed

- Improved 3D color shading, and fixed some cases with incorrect lighting.
- Fixed some scenarios where GPU memory wasn't freed when it could have been freed.
- WebGL context wasn't marked as 'lost' when chart/dashboard was completely disposed.
- Fixed `GL_INVALID_VALUE : glScissor`` warning.
- Fixed crash when Pie chart of type `PieChartTypes.LabelsOnSides`` was created with only one slice.
- Fixed 3D Point Line Series point size not updated to match highlight size when highlighted.
- Fixed 3D Point Line Series point size could be smaller than line thickness, after changing line thickness, resulting in visible gaps in line.

### ### Deprecated

- `Theme.seriesNonTriangulatedPointStyle3D``. Use `Theme.pointCloudSeries3DPointStyle`` instead.
- `Theme.seriesTriangulatedPointStyle3D``. Use `Theme.pointSeries3DPointStyle`` instead.
- `Theme.customTickMarkerFillStyle``. Use `Theme.uiPointableTextBoxFillStyle`` instead.
- `Theme.customTickMarkerTextFillStyle``. Use either `customTickMarkerTextFillStyle`` or `uiTickTextFillStyle``.
- `Theme.customTickMarkerStrokeStyle``. Use either `uiPointableTextBoxStrokeStyle`` or `uiTickStrokeStyle``.
- `Theme.pointMarkerHorizontalGridStrokeStyle``. Use `customTickGridStrokeStyle`` instead.
- `Theme.pointMarkerVerticalGridStrokeStyle``. Use `customTickGridStrokeStyle`` instead.
- `Theme.axisLabelFillStyle``. Use `numericTickStrategy`` or other `_tick strategy_`` property instead.
- `Theme.axisLabelFont``. Use `numericTickStrategy`` or other `_tick strategy_`` property instead.
- `Theme.pointMarkerTextFillStyle``. Use `uiPointableTextBoxTextFillStyle`` or `uiTickTextFillStyle`` instead.
- `Theme.seriesStrokeStyle3D``. Use `Theme.lineSeries3DStrokeStyle`` instead.
- `Theme.seriesNonTriangulatedPointStyle3D``. Use `Theme.pointCloudSeries3DPointStyle`` instead.
- `Theme.seriesTriangulatedPointStyle3D``. Use `Theme.pointSeries3DPointStyle`` instead.



- `LineSeries3D.setStyle``. Use `setStrokeStyle`` instead.
- `LineSeries3D.getLineStyle``. Use `getStrokeStyle`` instead.
- `LineSeries3D.setStyleHighlight``. Use `setStrokeStyleHighlight`` instead.
- `LineSeries3D.getLineStyleHighlight``. Use `getStrokeStyleHighlight`` instead.
- `PointLineSeries3D.setStyle``. Use `setStrokeStyle`` instead.
- `PointLineSeries3D.getLineStyle``. Use `getStrokeStyle`` instead.
- `PointLineSeries3D.setStyleHighlight``. Use `setStrokeStyleHighlight`` instead.
- `PointLineSeries3D.getLineStyleHighlight``. Use `getStrokeStyleHighlight`` instead.
- `ChartXY.disableAnimations()`` use `ChartXY.setAnimationsEnabled(false)`` instead.
- `Axis.disableAnimations()`` use `Axis.setAnimationsEnabled(false)`` instead.
- `Dashboard.disableAnimations()`` use `Dashboard.setAnimationsEnabled(false)`` instead.
- `Spider.disableAnimations()`` use `Spider.setAnimationsEnabled(false)`` instead.
- `Polar.disableAnimations()`` use `Polar.setAnimationsEnabled(false)`` instead.
- `Pie.disableAnimations()`` use `Pie.setAnimationsEnabled(false)`` instead.
- `Gauge.disableAnimations()`` use `Gauge.setAnimationsEnabled(false)`` instead.
- `Funnel.disableAnimations()`` use `Funnel.setAnimationsEnabled(false)`` instead.
- `Pyramid.disableAnimations()`` use `Pyramid.setAnimationsEnabled(false)`` instead.
- `Map.disableAnimations()`` use `Map.setAnimationsEnabled(false)`` instead.
- `Chart3D.disableAnimations()`` use `Chart3D.setAnimationsEnabled(false)`` instead.
- `Series.setMaxPointCount()`` use `Series.setDataCleaning()`` instead.
- `Series.setDataCleaningThreshold()`` use `Series.setDataCleaning()`` instead.

## [3.0.1] - 2021-05-26

### ### Added

- `PointSeries`, `PointLineSeries`, `SplineSeries` and `StepSeries` now support data point 'value' paletted coloring.

### ### Fixed

- `PointSeries`, `PointLineSeries`, `SplineSeries` and `StepSeries` now properly interact with Legend LUT UI element if styled with `PalettedFill`.
- `LineSeries` and `AreaSeries` stroke gradient stroke style sometimes would render incorrectly.
- WebGL Errors in some scenarios when running 3D charts on Android Chrome.
- Fixed LUT Unit label clipping out of legend box in some configurations.
- Rectangle gradient stroke style not working properly.
- Chart would try to zoom out when zooming in with touch gesture.
- Surface 3D theme style only applied after the series was highlighted or style was explicitly set.
- Box Series 3D legend box entry not styled according to the series fill style.
- Axis pan direction could be reversed when chart was in dashboard and the chart was resized using a splitter and the axis was reversed.
- Axis interval selection visual incorrect height in some scenarios.
- Opposite axis positioning was incorrect before first update after the axis was added.
- Polar Sector low resolution in some cases.
- Fixed missing `LineSeries` mouse interactions.
  - `LineSeries` Mouse interactions are disabled by default for performance reasons regarding freeform line series.
  - Call `LineSeries.setMouseInteractions(true)` to enable mouse interactions if interactions are needed.
- Fixed minor grid lines sometimes rendering over major grid lines



 **[3.0.0] - 2020-05-05****### Added**

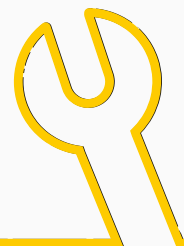
- PolarChart
- PolarAxisAmplitude
- PolarAxisRadial
- PolarPoint
- PolarPointSeries
- PolarLineSeries
- PolarPointLineSeries
- PolarAreaSeries
- PolarPolygonSeries
- PolarSector
- Logarithmic axis support for XY charts
- BoxSeries3D now supports PalettedFill by x, y or z
- LineSeries.setDataCleaningThreshold, getDataCleaningThreshold
- LineSeries.setCursorSolveBasis, getCursorSolveBasis
- PointLineSeries.setCursorSolveBasis, getCursorSolveBasis
- SplineSeries.setCursorSolveBasis, getCursorSolveBasis
- StepSeries.setCursorSolveBasis, getCursorSolveBasis
- UILUTCheckBox
- API for Axis mouse and touch events
- UITick
- UITickBuilder
- UIElementBuilders.AxisTick
- Configuration options for changing mouse interactions to different mouse buttons
- UILegendBoxPanel.setLegendBoxes
- NumericTickStrategy.setExtremeFormattingFunction
- NumericTickStrategy.setMajorFormattingFunction
- NumericTickStrategy.setMinorFormattingFunction
- API for subscribing to mouse and touch events on chart background
- Dependency to earcut (<https://github.com/mapbox/earcut>).  
Used for Polygon triangulation.





### ### Changed

- Attaching a series with color lookup table (LUT) now automatically visualizes the color steps with a LUTUICheckBox component.
- LegendBox title is no longer automatically set to match chart title. Title can be set with new method LegendBox.setTitle.
- Tweaked LegendBox default style to look a bit nicer.
- LegendBox.add API has changed. Refer to migration guide for details.
- LegendBoxEntry is now styled accordingly with series PalettedFill.
- Nib mouse wheel behavior is now more intuitive
- Default style of XY Markers was changed to same as AutoCursor
- Renamed CustomTick.setTopPadding -> setTickLabelPadding
- Renamed PointableTextBox -> UIPointableTextBox
- Major improvements to text rendering performance
- Changed default CheckBox button picture to UIButtonPictures.Circle
- Renamed ResultTable.setFont to setTextFont
- Renamed UITextBox.setFont to setTextFont
- Renamed UICheckBox.setFont to setTextFont
- Renamed LegendBoxEntry.setFont to setTextFont
- Theme.chartBackgroundFillStyle renamed to seriesBackgroundFillStyle
- Theme.chartBackgroundStrokeStyle renamed to seriesBackgroundStrokeStyle
- on/offChartBackground... event methods were renamed to on/offSeriesBackground...
- setChartBackgroundFillStyle methods renamed to setSeriesBackgroundFillStyle
- getChartBackgroundFillStyle methods renamed to getSeriesBackgroundFillStyle
- setChartBackgroundStrokeStyle methods renamed to setSeriesBackgroundStrokeStyle
- getChartBackgroundStrokeStyle methods renamed to getSeriesBackgroundStrokeStyle
- Changed default CheckBox button picture to UIButtonPictures.Circle
- UILegendBoxPanel.add no longer accepts series, or other attachables. Only chart or dashboard can be supplied.
- Default LegendBox alignment changed from horizontal to vertical.
- Improved default LegendBox positioning.
- Changed default cursor solve basis of all line series' to 'nearest-x'. This can be changed with new method: setCursorSolveBasis
- Improved Line Series rendering algorithms for progressive and real-time rendering, to perform much faster and look better.
- Heavily optimized progressive Line Series with user zooming in/out and automatic scrolling
- Significantly improved Line Series memory usage in scrolling applications with data cleaning enabled
- AreaSeries and AreaRangeSeries cursor now behaves as expected, by picking closest data point along X dimension. Cursor now also performs better.
- All dataPattern options have been changed. Instead of selecting an option from DataPatterns export, use object format instead, for example { pattern: 'ProgressiveX' }. See documentation/migration guide for details.
- onPanelBackground... -methods were renamed to onBackground...
- Renamed setResultTableFormatter, and getResultTableFormatter methods to setCursorResultTableFormatter, and getCursorResultTableFormatter respectively.
- Improved API documentation



### ### Removed

- `DefaultLibraryStyle` - Use `Themes.dark` or any other Theme instead.
- `UILegendBoxPanel` title API (`setTitle`, `getTitle`, `setTitleFillStyle`, `getTitleFillStyle`, `setTitleFont`, `getTitleFont`).
- `UILegendBoxPanel.setEntries`, use `UILegendBoxPanel.setLegendBoxes` for same functionality.
- Boolean parameter to `ChartXY.addAxisX` and `addAxisY`. Replaced with object syntax
- `NumericAxisTickStrategy.setFormattingFunction`. Replaced with individual setters for each tick level.
- `CustomTick.setPaddingBottom` (can be accessed via `TickMarker` background)
- `CustomTick.setSidePaddings` (can be accessed via `TickMarker` background)
- `CursorBuilderXY.setTickMarkerXBackground`
- `CursorBuilderXY.setTickMarkerYBackground`
- `PointSeriesOptions3D.pointShape`, use `PointSeries3D.setPointStyle` instead.
- `PointLineSeriesOptions3D.pointShape`, use `PointLineSeries3D.setPointStyle` instead.
- `Theme.numericTickStrategy3D` use `Theme.numericTickStrategy` instead.
- `Theme.dateTimeTickStrategy3D` use `Theme.dateTimeTickStrategy` instead.
- `DataPatterns`. Use object format instead, for example `{ pattern: 'ProgressiveX' }`. See [documentation/migration guide](#) for details.

### ### Fixed

- Fixed ticks overlapping other ticks in some cases
- Fixed LUT unexpected color step behavior when interpolation is disabled
- Fixed `Axis.fit()` when Series points are in a straight line
- Fixed progressive Axis scrolling sometimes getting ahead of series
- Fixed scenario where chart rendered with 1px x 1px canvas
- Fixed Constant line and Band touch events not working inside dashboard.
- Fixed mouse leave event not fired correctly in all cases
- Fixed missing configuration for 3D chart creation inside dashboard, `theme` and `disableAnimations` can now be properly set
- Fixed improper dispose behavior of `ChartXY`
- `IntensitySeries` now supports highlighting when styled with `SolidFill`
- `IntensitySeries` style is now properly matched in `LegendBox`
- `SurfaceSeries3D` style is now properly matched in `LegendBox`
- `SurfaceSeries3D` is now properly highlighted when hovering over respective `LegendBoxEntry`
- Fixed some cases where series boundaries were one frame behind actual boundaries
- Resolved some Z-fighting issues with `SurfaceSeries3D` wireframe and surface.
- Fixed `SurfaceSeries3D` and `HeatmapMeshSeries` not accounting boundaries of first and last column & row.
- Fixed axis nib mouse wheel scroll event wasn't properly stopped when interaction happened.
- Fixed `IntensitySeries` not updating when using `addColumn` to add only values
- Fixed type issues when using strict type checking mode with TypeScript
- Fixed linear gradient interpolation being incorrect when `devicePixelRatio` wasn't exactly 1.



## [2.2.1] - 2020-01-28

### ### Fixed

- License verification error when using a valid license

## [2.2.0] - 2020-01-27

### ### Added

- Chart3D series background
  - Chart3D.setSeriesBackgroundFillStyle
  - Chart3D.setSeriesBackgroundStrokeStyle
- 3D BoundingBox style API
  - Chart3D.setBoundingBoxStrokeStyle
  - Chart3D.getBoundingBoxStrokeStyle
- 3D camera behavior control
  - Chart3D.setCameraAutomaticFittingEnabled
  - Chart3D.getCameraAutomaticFittingEnabled
- New color palettes for palettes that match with the themes added in 2.1.0
  - auroraBorealis
  - blueSciFi
  - light
  - monochrome
  - night
  - sunset2
- Add get/setResultTableFormatter for IntensitySeries
- Intensity series wireframe. This was included in 2.1.0 release but was missing from the changelog. Changelog for 2.1.0 has also been updated to include this change.

### ### Changed

- Major improvements to 3D Point Series and 3D Line Series performance
- Improved 3D Box Series performance
- Improved Axis3D tick and grid line rendering performance
- General performance improvements
- Tweaked Axis3D joint style with thick line style
- Inconsistent RangeSeries max point count value to be consistent (0) with other series types.
- Improved default Chart3D camera behavior to fit data into visible viewport better
- Improved default themes paletted color behavior
- Improved 3D Line Series visual style
- Added 'shape' option to PointStyle3D.Triangulated for changing 3D points shape during runtime.
- Tweaked themes



### ### Fixed

- 3D theme inconsistencies
- `Axis.setScrollStrategy` TS type not accepting `undefined`
- `IntensitySeries` zooming and panning did unnecessary calculation

### ### Deprecated

- `Theme.numericTickStrategy3D` use `Theme.numericTickStrategy` instead.
- `Theme.dateTimeTickStrategy3D` use `Theme.dateTimeTickStrategy` instead.
- `ChartXY.setChartBackgroundFillStyle` renamed to `setSeriesBackgroundFillStyle`
- `ChartXY.setChartBackgroundStrokeStyle` renamed to `setSeriesBackgroundStrokeStyle`
- `PointSeries3D` 'pointShape' initialization configuration. Use `PointSeries3D.setPointStyle` instead.
- `PointLineSeries3D` 'pointShape' initialization configuration. Use `PointLineSeries3D.setPointStyle` instead.

## [2.1.0] - 2020-11-30

### ### Added

- 3D Surface Series
- 3D Box Series
- Added 3D Tick gridlines.
- Heatmap cursor support
- Custom Theme API to create your own Themes.
  - `customSimpleTheme()`
  - `customComplexTheme()`
  - `customTheme()`
- New themes
  - `AuroraBorealis`
  - `BlueSciFi`
  - `Classy`
  - `DarkGradient`
  - `Lavender`
  - `LavenderGradient`
  - `LightGradient`
  - `Lipstick`
  - `Monochrome`
  - `Night`
  - `Raspberry`
  - `Sunset`
- `Series.onHighlight` / `Series.offHighlight`
- `Axis.getInterval()` method for retrieving the currently applied axis scale interval.
- Added `vec3utils`. A collection of 3-dimensional vector math functions.
- Intensity series wireframe



### ### Changed

- Highlighting a Series on a Chart also highlights corresponding Series in attached Zoom Band Chart

### ### Fixed

- Fix not being able to style 3D Tick lines.
- Fixed error when running the charts in Node JS. (With the lcjs-headless package)
- Pie, Funnel and Spider charts animation speed inconsistency with ChartXY animation speed.
- Rendering error on first frame.

### ### Deprecated

- `DefaultLibraryStyle` - Use `Themes.dark` or any other Theme instead.

## [2.0.3] - 2020-10-15

### ### Fixed

- Rendering error after chart with PointSeries had been disposed once and recreated with a new PointSeries.

## [2.0.2] - 2020-09-17

### ### Fixed

- Fixed issue with PointSeries in Mac systems.
- Fixed AutoCursor being out of sync with actual position.
- Fixed issues with AngularJS related to typings.

## [2.0.1] - 2020-09-07

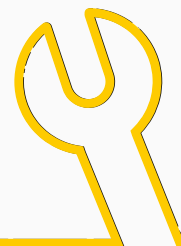
### ### Fixed

- Fixed new example links in readme file

## [2.0.0] - 2020-09-04

### ### Added

- Add public API to get Highlighters for Axis
- Added `onPositionChange` event to `ChartMarker` and `SeriesMarker`
- Added more mouse and touch events to chart background
- `layout()` method to `PublicEngine` interface.
- Add `LinearGradientFill` and `RadialGradientFill` fill styles.
- Added Axis `setTickStrategy` method.
- Added `AxisTickStrategies.Empty` (replaces `Axis.setTickStyle( emptyTick )`)
- Added `VisibleTicks.setLabelAlignment()`
- `Chart3D`
- `Axis3D`
- `Point3D`
- `PointSeries3D`
- `PointCloudSeries3D`
- `PointStyle3D`
- `LineSeries3D`
- `PointLineSeries3D`



### ### Changed

- Refactored thick line rendering.
  - Improves the visual style of the line when a semi-transparent fill style is used.
  - Improves GL memory usage. Same line now takes 40 to 70% less GL memory depending on the line.
- PointSeries now uses faster rendering methods on supported devices.
  - Major performance improvement
  - Uses less GL memory
- columnSpan and rowSpan Dashboard options are now optional and default to 1.
- Chart options when creating a chart inside dashboard are simplified.
  - Options are no longer behind a separate object but are instead given in same object as the dashboard options.
- Reduced zooming animation duration
- Made transparent backgrounds possible.
- LightningChart JS now requires the following WebGL extensions to work properly
  - ANGLE\_instanced\_arrays
  - EXT\_blend\_minmax
  - OES\_element\_index\_uint
  - OES\_standard\_derivatives
  - OES\_vertex\_array\_object
  - WEBGL\_lose\_context
  - If any of these extensions is missing, then a dismissible warning will be shown to notify users of possibly incorrectly working features.
- Changed seriesBackground mouse and touch event naming to chartBackground mouse and touch events.
  - e.g. onSeriesBackgroundMouseDown was changed to onChartBackgroundMouseDown
- requestAnimationFrame and cancelAnimationFrame are no longer polyfilled automatically.
  - The methods should be polyfilled if support for environments where the methods don't exist is needed.
- Numeric Axis ticks have been reworked
- DateTime Axis ticks have been reworked
- Mouse and touch events are properly handled to allow normal browser interactions when chart does no action for the event
- Changed ColorHEX factory color syntax to follow CSS color syntax, #RRGGBB[AA]
- Changed default highlight behavior of Series.
  - Series is no longer highlighted by default when hovering over it.
  - Use `series.setHighlightOnHover()` or `chart.setSeriesHighlightOnHover()` to highlight on hover.
- Improved styles for default themes.

### ### Removed

- AxisTickStrategies.NumericWithUnits
- Passing AxisTickStrategy of default Axes upon creating a ChartXY
- Passing AxisTickStrategy as parameter of ChartXY.addAxisX() or ChartXY.addAxisY()
- Removed APIs that were previously marked deprecated.
  - `Chart.setDataLabelFormater`
  - `Chart.getDataLabelFormater`
  - `Chart.setChartBackgroundStroke`
  - `Chart.getChartBackgroundStroke`
  - `Series.setMaxPointsCount`
  - `containerId` option

### ### Fixed

- Fix legend box item checkbox looking bad when stroke style is other than 1.
- Fix touch events triggering outside chart area



## [1.3.1] - 2020-05-29

### ### Fixed

- Inconsistent cursor style changing when moving from hovering element to hovering one element to hovering over another element.
- Fixed a crash when adding data to OHLCSeries
- Hovering over the Arction logo for long enough caused the chart to freeze itself instead of allowing clicking a link to go to the Arction website.
- Incorrect resolution and interactions when opening a chart in fullscreen mode

## [1.3.0] - 2020-04-28

### ### Added

- Heatmap
- `ChartXY.addHeatmapSeries()`
- `IntensityGridSeries` can be used for visualization of magnitude in two dimensions.
- `IntensityMeshSeries` can be used for visualization of magnitude in two dimensions, where the geometry of the series can be edited.
- `Dashboard.createZoomBandChart()`
- Added Axis Bands and ConstantLines.
- Application/Intranet Deployment key support
- A way to disable all animations at once.
  - Call `disableAnimations()` on any chart or specify `disableAnimations: true` as a chart creation option.
- Added `series.addArrayX()`, `.addArrayY()`, `.addArrayXY()` to basic Series types in XY Charts for user convenience.
  - These methods cause some overhead when used, so using the `series.add()` is still recommended for best performance.
- Added `.addArrayY()` to `OHLCSeriesWithAutomaticPacking` for user convenience.
  - This method causes some overhead when used, using `.add()` method is still recommended for best performance.
- `getSeries()` method to XY Charts and Spider chart.
- Support for rendering in Node JS environment with the help of "@arction/lcjs-headless" package.
- `renderFrame()` method to engine.
- Added `getDataLabelFillStyle` and `setDataLabelFillStyle` to `SolidGauge`.

### ### Changed

- `FitEngineToDiv.container` to pass DOM Element to Engine. Users can pass either DOM Element itself, or its ID.
- Mouse and touch interaction handling to add support for pen and PC touch screen interactions.
  - All interactions still work the same way, interactions just has better support for more interaction methods.

### ### Fixed

- Rendering error on some GPU's where GPU received some vertices as NaN instead of a valid vertex.
- Inconsistent style for `SeriesMarker` tick X and Y labels
- Fixed a `TypeError` on pyramid chart
- `LightningChart JS` logo failing to render correctly when high-dpi mode is used and `devicePixelRatio` is less than 1
- `LightningChart JS` logo duplication in `LegendBoxPanel`
- Incorrect cursor styles when hovering over axis or other element with resize cursor style
- `PointSeries.add` method requiring added points to be of `ColorPoint` type when `Point` is a valid type for it.

### ### Deprecated

- Deprecated use of `FitEngineToDiv.containerId` in `EngineOptions`. Use `FitEngineToDiv.container` instead.
- Deprecated use of `setChartBackgroundStroke` in `SpiderChart`, `ChartXY`. Use `setChartBackgroundStrokeStyle` instead.
- Deprecated use of `getChartBackgroundStroke` in `SpiderChart`, `ChartXY`. Use `getChartBackgroundStrokeStyle` instead.



## [1.2.2] - 2020-01-07

### ### Changed

- Error message when trying to create a chart in container that doesn't exist

### ### Fixed

- Mouse interactions permanently disabled when interactions disabled while interaction is in-progress
- Multiple Slice Explosion restriction not always working

## [1.2.1] - 2019-12-18

### ### Fixed

- TypeScript typings duplicate identifier

## [1.2.0] - 2019-12-10

### ### Added

- Added Axis Bands and ConstantLines.
- ColorHEX supports format with '0x' prefix.
- Dashboard.setSplitterStyle()
- Dashboard.setSplitterStyleHighlight()
- Dashboard.setBackgroundFillStyle()
- Dashboard.setBackgroundStrokeStyle()
- High DPI rendering support with 'devicePixelRatio' engine option
- Themes, with 'dark' and 'light' preset themes available.
- LUT (ValueRangePalette)
- series.getPointAmount() in XY Charts
- ChartXY.setMouseInteractionsWhileScrolling()
- ChartXY.setMouseInteractionsWhileZooming()
- New option when creating lines: highlightThicknessMultiplier can be used to specify thickness of highlighted lines
- Progressive DataPatterns precision

### ### Changed

- Chart examples linked in the readme
- Mouse interactions are disabled by default when scrolling / zooming. This behavior can be changed with methods in XY Charts.

### ### Fixed

- GL errors with Pyramid Chart
- AutoCursor working incorrectly with touch displays





## [1.1.1] - 2019-10-11

### ### Changed

- Chart examples linked in the readme

### ### Fixed

- Typo in the readme
- Missing typings
- Performance issue found in the trading showcase

## [1.1.0] - 2019-10-03

### ### Added

- Dispose API for Charts, Dashboard
- Add `OHLCSeries.set/getFigureAutofitting()`

### ### Changed

- Removed dependency to `crc` package.
- Removed dependency to `collections` package.

### ### Deprecated

- `SolidGauge.setDataLabelFormatter` in favor of `SolidGauge.setDataLabelFormatter`
- `SolidGauge.getDataLabelFormatter` in favor of `SolidGauge.getDataLabelFormatter`
- `OHLCSeries.setMaxPointsCount` in favor of `OHLCSeries.setMaxPointCount`

### ### Fixed

- Fixed crash when adding points to step series.
- `PointableTextBox` incompatibility with axis custom tick.
- Empty `StrokeStyle` not working with `Axis Nibs`

## [1.0.3] - 2019-08-26

### ### Added

- More keywords for `npm`

### ### Changed

- Readme file contents present the package better

## [1.0.1] - 2019-08-14

### ### Added

- `CHANGELOG.md` included in the `npm` package
- More keywords for `npm`

### ### Fixed

- Pyramid Chart
  - Added missing API documentation for `get/setAnimationsEnabled`
- Funnel Chart
  - Added missing API documentation for `get/setAnimationsEnabled`



## [1.0.0] - 2019-08-05

### ### Added

- 2-Dimensional Charts
  - ChartXY (Cartesian Chart)
    - Line Series
    - Point Series
    - Point Line Series
    - Spline Series
    - Step Series
    - Rectangle Series
    - Ellipse Series
    - Box Series
    - OHLC Series
    - Area Series (BiPolar, Monopolar, AreaRange)
    - Axes
  - Spider Chart
  - Pie Chart
  - Gauge Chart
  - Funnel Chart
  - Pyramid Chart
- Markers (for Series / Chart)
- Cursor
- Mouse Interactions
- Touch Support
- Animations
- Dashboard
- LegendBox
- UI Elements (Buttons, CheckBoxes, Labels, TextBoxes)



## [3.2.0] - 2021-09-22

### ### Added

- Added `DateTickStrategy.setCursorFormatter`` for easy modification of cursor ticks formatting.
- Added `NumericTickStrategy.setCursorFormatter`` for easy modification of cursor ticks formatting.
- Added `TimeTickStrategy.setCursorFormatter`` for modification of cursor ticks formatting.
- Added `axis.setThickness``.
  - This can be used to control the width or height of axis depending on is the axis Y or X axis.
- Added primitive draw mode to `LineSeries3D`` and `PointLineSeries3D``.
  - This feature can be used for improved performance by reducing line depth perception.
  - See `setStrokeStyle`` method for more information.
- Added `series.setAutoScrollingEnabled`` for disabling scrolling and fitting of specific series.
- Added `synchronizeAxisIntervals``. Utility for synchronizing intervals of one or more axis.

### ### Changed

- Improved default Date time cursor formatting to display Year, month, day, hour and minute information. This can be modified with new method `DateTickStrategy.setCursorFormatter``.
- Added `dataPoint`` parameter to `SeriesXYFormatter``. This allows using `value``, `size``, `color``, `rotation`` and custom values in cursor result table formatters.
- `ZoomBandChart`` now automatically aligns itself with the reference chart.
- Community version no longer has performance penalty.
- `NumericTickStrategy.setFormattingFunction`` now also affects default cursor result table formatting and cursor ticks formatting.
- `SolidGauge.setIntervalLabelFormatter`` and `setDataLabelFormatter`` now also accept callback functions which can be used to display text that is not strictly numbers.
- Improved axis tick rendering performance.

### ### Fixed

- Fixed ChartXY UI element positioning on axis.
- Fixed map chart AutoCursor not usable with touch.
- Fixed `ZoomBandChart`` not reacting to `Dashboard`` `theme`` or `disableAnimations`` options.
- `PointSeries3D`` no longer crashes when styled with `PalettedFill``.
- `PointSeries3D`` now displays color lookup table range when attached to a legend box and styled with `paletted fill``.
- Fixed edge case in text rendering when creating text with large font size.
- Fixed occasional error message when zooming in and out line series rapidly.
- Fixed incorrect rendering behavior when chart is transitioned to fullscreen on Safari.
- Fixed incorrect legend box entry alignment after restoring the legend box entry.
- Fixed `OHLCSeries`` sometimes not being visible, especially when chart is created and animations are disabled.
- Fixed point, line and range series `getPointAmount`` not counting points that were added just before.
- Fixed empty series affecting automatic fitting of series.
- LightningChart JS logo now renders properly on high DPI devices with `window.devicePixelRatio`` greater than 1.
- Fixed poor performance when adding new points to charts with 1 million or more points in some cases.
- Fixed `HeatmapGridSeries.getWireframeStyle`` returning `FillStyle`` instead of `LineStyle``.
- Fixed TypeScript issue with being able to pass `LineStyle`` as `FillStyle`` or vice versa.
- Fixed `HeatmapScrollingGridSeries`` not attaching to any other axis than default chart axis.
- Fixed `IntensityGridSeries`` and `IntensityMeshSeries`` not rendering after it has been restored.

