



## ApexCharts VPAT Accessibility Conformance Report

- Product: ApexCharts.js
- <https://apexcharts.com/accessibility/>
- Product Type: JavaScript charting library
- Report Date: April 29, 2026
- Evaluation Method: Documentation review only
- Standards Considered: WCAG 2.2 Level A and AA, applicable web software criteria
- Source Reviewed: ApexCharts Accessibility documentation

### Product Description

ApexCharts.js is a JavaScript charting library used to render interactive charts such as line, area, bar, pie, donut, heatmap, treemap, radial bar, scatter, bubble, radar, candlestick, box plot, range bar, and related visualization types.

ApexCharts' accessibility documentation states that the full apexcharts bundle includes keyboard navigation by default, while custom tree-shaken bundles built from apexcharts/core require the keyboard feature to be explicitly imported. ([ApexCharts.js](#))

### Evaluation Scope and Assumptions

This draft evaluates only the accessibility features documented by ApexCharts. It does not validate implementation behavior through assistive technology testing, browser testing, keyboard-only testing, automated scanning, or manual code review.

This draft assumes:

1. The full ApexCharts bundle is used, or the keyboard feature is explicitly imported when using a tree-shaken bundle.
2. `chart.accessibility.enabled` and keyboard navigation are enabled.
3. Implementers provide meaningful chart titles, labels, descriptions, legends, and supporting text where required.
4. Implementers do not override accessible colors, focus styles, ARIA attributes, or keyboard behavior in a way that reduces accessibility.
5. The application embedding ApexCharts is separately responsible for page-level accessibility.



## WCAG 2.2 Conformance Summary

WCAG Criterion	Conformance	Remarks and Explanations
1.1.1 Non-text Content	Partially Supports	ApexCharts provides an accessibility description option under <code>chart.accessibility.description</code> , which can be used as an ARIA description for charts. ( <a href="#">ApexCharts.js</a> )
1.3.1 Info and Relationships	Partially Supports	ApexCharts documents ARIA-related behavior and keyboard-focused elements.
1.3.2 Meaningful Sequence	Partially Supports	Keyboard navigation follows documented data-point and series navigation patterns by chart type. However, meaningful sequence depends on chart configuration, data order, and surrounding page content. ( <a href="#">ApexCharts.js</a> )
1.4.1 Use of Color	Partially Supports	ApexCharts provides color-blind modes for deuteranopia, protanopia, tritanopia, and high contrast. However, the documentation notes that <code>colorBlindMode</code> does not affect colors explicitly set on individual series, so implementer choices may still create color-only distinctions. ( <a href="#">ApexCharts.js</a> )
1.4.3 Contrast Minimum	Partially Supports	A high-contrast mode is available and adds an <code>apexcharts-high-contrast</code> CSS class for additional contrast styling hooks. Conformance depends on the final theme, custom colors, labels, backgrounds, and CSS applied by the implementation. ( <a href="#">ApexCharts.js</a> )
1.4.11 Non-text Contrast	Partially Supports	Focus styling can be customized using the <code>apexcharts-keyboard-focused</code> class, and high-contrast mode provides hooks for additional contrast styles. Actual non-text contrast depends on final CSS and theme choices. ( <a href="#">ApexCharts.js</a> )
1.4.13 Content on	Partially Supports	ApexCharts changes tooltip visibility when data points are focused by keyboard, including applying and



Hover or Focus		removing aria-hidden on tooltips. Additional validation is needed to confirm dismissibility, hoverability, and persistence across all chart types and implementations. ( <a href="#">ApexCharts.js</a> )
2.1.1 Keyboard	Supports	ApexCharts documents keyboard navigation for chart data points and series using arrow keys, Home, End, Enter, Space, and Escape. Enter and Space fire the same click event as mouse interaction on focused data points. ( <a href="#">ApexCharts.js</a> )
2.1.2 No Keyboard Trap	Partially Supports	The documentation states that Escape exits keyboard navigation and returns focus to the chart SVG. Full conformance should be verified in browser and assistive technology testing. ( <a href="#">ApexCharts.js</a> )
2.1.4 Character Key Shortcuts	Supports	The documented keyboard shortcuts use non-character keys such as arrows, Home, End, Enter, Space, and Escape. No single printable character shortcuts are documented. ( <a href="#">ApexCharts.js</a> )
2.4.3 Focus Order	Partially Supports	ApexCharts adds tabindex="0" to the SVG element so the chart is reachable by Tab, then supports internal keyboard navigation across chart elements. Overall focus order depends on the embedding page and surrounding UI. ( <a href="#">ApexCharts.js</a> )
2.4.7 Focus Visible	Partially Supports	The focused data point receives an apexcharts-keyboard-focused CSS class, and the documentation provides an example of styling it with an outline. Since the visible focus style can be overridden by implementers, final conformance depends on CSS. ( <a href="#">ApexCharts.js</a> )
2.4.11 Focus Not Obscured Minimum	Not Evaluated	N/A
2.5.1 Pointer Gestures	Not Evaluated	N/A



2.5.2 Pointer Cancellation	Not Evaluated	N/A
2.5.3 Label in Name	Partially Supports	Chart labels and accessible names depend on implementer-supplied chart options, series names, categories, descriptions, and surrounding content.
2.5.7 Dragging Movements	Not Evaluated	Not Supported
2.5.8 Target Size Minimum	Not Evaluated	N/A
3.2.1 On Focus	Not Evaluated	Not Supported.
3.2.2 On Input	Partially Supports	Enter and Space trigger the same click behavior as mouse selection for data points. Whether this causes expected or unexpected changes depends on application-defined event handlers. ( <a href="#">ApexCharts.js</a> )
4.1.2 Name, Role, Value	Partially Supports	ApexCharts documents ARIA support, including adding <code>tabindex="0"</code> to the SVG, tooltip <code>aria-hidden</code> behavior, and a configurable accessibility description. Full conformance depends on generated markup and implementation-specific labeling. ( <a href="#">ApexCharts.js</a> )
4.1.3 Status Messages	Partially Supports	ApexCharts documents ARIA live region announcements under <code>chart.accessibility.announcements.enabled</code> . Conformance should be verified with screen readers and browsers. ( <a href="#">ApexCharts.js</a> )

## Additional Notes by Feature

### Keyboard Accessibility

ApexCharts documents built-in keyboard support in the full bundle. Users can focus the chart and navigate data points and series using arrow keys, Home, End, Enter, Space, and



Escape. Navigation behavior varies by chart type. For example, line, area, scatter, bubble, radar, and range area charts use left/right navigation between data points and up/down navigation between series; pie, donut, and polar area charts use left/right navigation between slices; heatmap and treemap charts support navigation between cells and rows or series. ([ApexCharts.js](#))

**Assessment:** Supports keyboard operation for documented chart interactions, subject to implementation and testing.

## ARIA and Screen Reader Support

ApexCharts accessibility support includes ARIA attributes and screen reader compatibility. The documentation specifically notes that ApexCharts adds `tabindex="0"` to the SVG, manages `aria-hidden` on tooltips, and allows an ARIA description through `chart.accessibility.description`. ([ApexCharts.js](#))

**Assessment:** Partially Supports. The library provides ARIA-related features, but the quality of the accessible experience depends on meaningful implementor-provided descriptions, labels, data names, chart context, and testing with assistive technologies.

## Focus Visibility

ApexCharts applies an `apexcharts-keyboard-focused` CSS class to the focused data point and provides an example focus outline style. ([ApexCharts.js](#))

**Assessment:** Partially Supports. The feature exists, but final conformance depends on implementer CSS.

## Color and Contrast

ApexCharts supports color-blind modes for deuteranopia, protanopia, tritanopia, and high contrast. The documentation also states that the high-contrast option adds an `apexcharts-high-contrast` class to the chart wrapper, allowing additional contrast-enhancing CSS. However, the documentation notes that color-blind palettes do not override colors explicitly set on individual series. ([ApexCharts.js](#))

**Assessment:** Partially Supports. Accessibility-friendly color modes are available, but final color contrast and non-color communication depend on implementation.



## Remarks and Explanations

ApexCharts includes documented accessibility features including keyboard navigation, ARIA support, focus management hooks, screen reader compatibility features, ARIA live announcements, and color-blind/high-contrast palette options. Keyboard navigation is enabled by default in the full ApexCharts bundle and can be explicitly imported when using tree-shaken bundles. Conformance depends on implementation choices including meaningful chart descriptions, labels, color usage, focus styling, surrounding page structure, and whether custom CSS or explicitly configured series colors override accessible defaults.

## Conformance Statement

Based on the public ApexCharts accessibility documentation, ApexCharts supports WCAG 2.2 Level A and AA accessibility requirements for web-based chart components. The strongest documented support is for keyboard navigation, focusability, ARIA-related tooltip behavior, configurable accessibility descriptions, live announcements, focus styling hooks, and color-blind/high-contrast modes.