IBM **Accessibility**

Equal Access Toolkit:

Accessibility Checker

Version 3.1.74

1. [How to install](chrome-extension://lkcagbfjnkomcinoddgooolagloogehp/usingAC.html#install)
2. [Accessibility issues](chrome-extension://lkcagbfjnkomcinoddgooolagloogehp/usingAC.html#issues)
3. [The Checker view](chrome-extension://lkcagbfjnkomcinoddgooolagloogehp/usingAC.html#view)
   * [3.1 Scanning](chrome-extension://lkcagbfjnkomcinoddgooolagloogehp/usingAC.html#a11y_check)
   * [3.2 Scan local files](chrome-extension://lkcagbfjnkomcinoddgooolagloogehp/usingAC.html#local_files)
   * [3.3 Synchronize views](chrome-extension://lkcagbfjnkomcinoddgooolagloogehp/usingAC.html#sync)
   * [3.4 Create a scan report](chrome-extension://lkcagbfjnkomcinoddgooolagloogehp/usingAC.html#t_single_scan_report)
   * [3.5 Create a multi-scan report](chrome-extension://lkcagbfjnkomcinoddgooolagloogehp/usingAC.html#t_multi_scan_report)
   * [3.6 Focus view](chrome-extension://lkcagbfjnkomcinoddgooolagloogehp/usingAC.html#focus_view)
   * [3.7 Filter views](chrome-extension://lkcagbfjnkomcinoddgooolagloogehp/usingAC.html#filter_views)
   * [3.8 Show/Hide issues](chrome-extension://lkcagbfjnkomcinoddgooolagloogehp/usingAC.html#hide_issues)
   * [3.9 Keyboard checker mode](chrome-extension://lkcagbfjnkomcinoddgooolagloogehp/usingAC.html#keyboard_checker_mode)
4. [The Assessment view](chrome-extension://lkcagbfjnkomcinoddgooolagloogehp/usingAC.html#a11y_assess)
5. [Settings](chrome-extension://lkcagbfjnkomcinoddgooolagloogehp/usingAC.html#t_select_settings)
6. [Checker reports](chrome-extension://lkcagbfjnkomcinoddgooolagloogehp/usingAC.html#the_report)
7. [Accessibility features](chrome-extension://lkcagbfjnkomcinoddgooolagloogehp/usingAC.html#a11y_considerations)
8. [Feedback](chrome-extension://lkcagbfjnkomcinoddgooolagloogehp/usingAC.html#feedback)
9. [Troubleshooting](chrome-extension://lkcagbfjnkomcinoddgooolagloogehp/usingAC.html#troubleshooting)

For bite-sized guidance, see the [Quick guide](chrome-extension://lkcagbfjnkomcinoddgooolagloogehp/quickGuideAC.html)

**User guide - IBM Accessibility Checker**

The IBM Equal Access Toolkit: Accessibility Checker ("**the Checker**") tests web pages for accessibility issues with W3C Web Content Accessibility Guidelines (WCAG 2.x), IBM Accessibility requirements, and other standards. The Checker is [also available as a Node package](https://www.ibm.com/able/toolkit/tools/#develop) for automated testing within a continuous integration / continuous delivery (CI/CD) pipeline. The Checker rules include explanations and help for suggested fixes.

* **Chrome browser extension**: integrates automated accessibility checking capabilities into the Chrome Developer Tools
* **Firefox browser add-ons**: integrates automated accessibility checking capabilities into the Firefox Web Developer Tools
* **Edge browser add-ons**: integrates automated accessibility checking capabilities into the Edge Developer Tools
* **Node accessibility-checker**: automated accessibility testing within a continuous integration pipeline, such as Travis CI for Node-based test environments, such as Selenium, Puppeteer, Playwright, and Zombie; the ability to validate results against baseline files and scan local files
* **Karma-accessibility-checker**: automated accessibility testing for the Karma environment
* **Cypress-accessibility-checker**: wrapper of the accessibility-checker in the Cypress environment

The CI/CD packages use the same engine and rules as the browser extensions, making it easy to replicate finding issues in either environment when the Rule set settings are similar.

**1. How to install**

Each supported browser comes with DevTools pre-installed, simplifying the installation and integration with the Checker. New users should become familiar with DevTools and its features and settings by visiting the documentation:

* Google Chrome and [Chrome DevTools](https://developer.chrome.com/docs/devtools/)
* Mozilla Firefox and [Firefox DevTools](https://firefox-source-docs.mozilla.org/devtools-user/)
* Microsoft Edge and [Edge DevTools](https://learn.microsoft.com/en-us/microsoft-edge/devtools-guide-chromium/landing/)

**Install for Chrome:**

1. Open the Chrome browser
2. Go to the [IBM Equal Access Accessibility Checker](https://chrome.google.com/webstore/detail/ibm-equal-access-accessib/lkcagbfjnkomcinoddgooolagloogehp) in the Chrome Web Store
3. Press 'Add To Chrome' button

**Install for Firefox:**

1. Open the Firefox browser
2. Go to the [IBM Equal Access Accessibility Checker](https://addons.mozilla.org/en-US/firefox/addon/accessibility-checker/) in Firefox Browser Add-ons
3. Press 'Add To Firefox' button

**Install for Edge:**

1. Open the Edge browser
2. Go to the [IBM Equal Access Accessibility Checker](https://microsoftedge.microsoft.com/addons/detail/ibm-equal-access-accessib/ompccpejakabkmfepbijnagedbdfldka) in Microsoft Edge Add-ons
3. Press 'Add To Edge' button

**DevTools layout and settings**

The Checker is a “**DevTools**” extension that includes an [Accessibility Checker](chrome-extension://lkcagbfjnkomcinoddgooolagloogehp/usingAC.html#view) tab to find and fix issues in the code and an [Accessibility Assessment](chrome-extension://lkcagbfjnkomcinoddgooolagloogehp/usingAC.html#a11y_assess) tab for an executive summary of the page. The DevTools settings affect the layout and appearance of the Checker:

1. **Dock side** settings affect the layout:
   * Dock to the right
   * Dock to the left
   * Dock to the bottom
2. **Appearance** settings are supported:
   * **Theme**: System preferences (recommended), Dark, and Light
   * **Panel Layout**: horizontal, vertical, and auto

For example, the screenshots in this User guide use 'Dock to the right', 'System preferences' with 'Dark' theme, and ‘horizontal’ panel layout. Experienced DevTools users will appreciate the power and functionality that the integration provides. All users may want to experiment with the 'Dock side' and 'Appearance' settings to take advantage of their own screen size and user preferences.

**2. Accessibility issues**

As with any automated test tool for accessibility, these tests don’t catch all issues. Complete the accessibility testing with [developer unit testing](https://www.ibm.com/able/toolkit/develop/considerations/unit-testing/) and follow all the [steps in the Verify phase](https://www.ibm.com/able/toolkit/verify) in the IBM Equal Access Toolkit. .

The issues are divided into three types:

* **Violation** - failures that need to be corrected
* **Needs review** - review to confirm that it's not a violation
* **Recommendation** - opportunities to apply best practices
* **Hidden** - issues to be ignored or have been resolved

There are three (3) ways to organize and group the set of issues detected:

* by **Element roles** – issues are organized by the ARIA roles of the Document Object Model (DOM) elements. This view shows both implicit and explicit roles, and not the element names. Use this view to explore issues within a specific element role and its children. (default)
* by **Requirements** – issues are mapped to the most relevant IBM requirement, which corresponds to the WCAG standards. Use this view to classify and report issues.
* by **Rules** - issues organized by rules in the rule set. Use this view to see all the issues common to a rule at once.

There are four (4) ways to further filter one or more **types** of issues detected:

* **Violations** – failures that need to be corrected (default checked)
* **Needs review** – review to confirm that it's not a violation (default checked)
* **RecommendationsRules** - opportunities to apply best practices (default checked)
* **Hidden** - issues to be ignored or have been resolved (default not-checked)

For in-depth guidance, view [Filter views](chrome-extension://lkcagbfjnkomcinoddgooolagloogehp/usingAC.html#filter_views) in the User guide.

**3. The Checker view**

The 'Accessibility Checker' tab is located in the 'Elements' panel in Chrome/Edge or the 'Inspector' panel in Firefox. Developers use this view to quickly find and fix issues in code and on the page.

Note: The location of the 'Accessibility Checker' tab is dependent on the layout and appearance settings in DevTools.

**To launch the Checker**, do one of the following:

* Launch Developer Tools:
  1. **Command+Option+I** on MacOS® or **Control+Shift+I** on Microsoft Windows®
  2. Or, Right-click the web page, select '**Inspect**'
  3. Select 'Elements' panel in Chrome/Edge or the 'Inspector' panel in Firefox
  4. Select 'Accessibility Checker' tab
* From the Chrome and Edge browser menu:
  1. Select 'More tools', then 'Developer Tools'
  2. Select 'Elements' panel
  3. Select 'Accessibility Checker' tab
* From the Firefox browser menu:
  1. Select 'More tools', then 'Web Developer Tools'
  2. Select 'Inspector' panel
  3. Select 'Accessibility Checker' tab

**3.1 Scan to find issues**

Press the blue 'Scan' button to quickly find issues with a component or web page.

Launch developer tools on the web page to be tested.

1. Open ‘Elements’ panel (Chrome/Edge) or ‘Inspector’ panel (Firefox)
2. Select 'Accessibility Checker' tab (usually in the lower or right-hand panel)
3. Press the ‘Scan’ button to get a list of issues

The scan results show the total number of issues grouped by the four (4) types:

* **Violation** - failures that need to be corrected
* **Needs review** - review to confirm that it's not a violation
* **Recommendation** - opportunities to apply best practices
* **Hidden** - issues to be ignored or have been resolved

There are three (3) ways to organize and group the set of issues listed:

* by **Element roles** – issues are organized by the ARIA roles of the Document Object Model (DOM) elements. This view shows both implicit and explicit roles, and not the element names. Use this view to explore issues within a specific element role and its children. (default)
* by **Requirements** – issues are mapped to the most relevant IBM requirement, which corresponds to the WCAG standards. Use this view to classify and report issues.
* by **Rules** - issues organized by rules in the rule set. Use this view to see all the issues common to a rule at once.

**Filter**: there are four (4) ways to further filter the set of issus listed by one or more **types** of issues detected:

* **Violations** – failures that need to be corrected (default checked)
* **Needs review** – review to confirm that it's not a violation (default checked)
* **RecommendationsRules** - opportunities to apply best practices (default checked)
* **Hidden** - issues to be ignored or have been resolved (default not-checked)

For in-depth guidance, view [Filter views](chrome-extension://lkcagbfjnkomcinoddgooolagloogehp/usingAC.html#filter_views) in the User guide.

**Additional actions** that can be taken with the list of issues:

* Collapse or Expand the group of issues to see the individual issues
* Select an individual or groups of issues to be hidden
* Click 'Learn more' link in the issue for detailed help on how to fix it
* Press 'Export to XLS' to download the results in a spreadsheet format

**Hidden content not scanned**

By default, the Checker skips hidden content (Web pages that use the visibility:hidden or display:none elements). When the content is changed or revealed to users at any point, re-scan the page. Trigger the dynamic content live in the browser and in test scripts and see [Create a multi-scan report](chrome-extension://lkcagbfjnkomcinoddgooolagloogehp/usingAC.html#t_multi_scan_report). Ensure the tests trigger the display of non-visible content so that the Checker can test it.

**3.2 Scan local files**

The Checker is able to scan local .html or .htm files launched in the Firefox browser by default. Follow the steps below to allow scanning of local .html or .htm files in the Chrome browser:

1. Open 'Extensions' from the Chrome browser toolbar
2. Open 'Manage Extensions' to see all installed extensions
3. Press the 'Details' button of the IBM Equal Access Accessibility Checker extension
4. Scroll down and turn on 'Allow access to file URLs'

**Automated batch scanning of local files**

The Checker is [available as a Node package](https://www.ibm.com/able/toolkit/tools/#develop) for automated testing within a continuous integration / continuous delivery (CI/CD) pipeline that can scan a batch of local files:

* **Node accessibility-checker**: automated accessibility testing within a continuous integration pipeline, such as Travis CI for Node-based test environments, such as Selenium, Puppeteer, Playwright, and Zombie; the ability to validate results against baseline files and scan local files
* **Karma-accessibility-checker**: automated accessibility testing for the Karma environment
* **Cypress-accessibility-checker**: wrapper of the accessibility-checker in the Cypress environment

**3.3 Synchronize views**

'Inspect' an element on the web page (or select an issue in the list of issues) to synchronize the highlighting on the web page, highlighting related issues in the issue list, and easily view the help information for the specific selected issue. Using the integrated power of DevTools, make changes to the code and re-scan to see the accessibility results.

Select an issue in the Checker, an element in the DOM, or 'Inspect' an element on the web page to:

* Show the number of issues for that selected element on the page
* Highlight all related issues, if any, with a light purple border (e.g., 4 issues in this example)
* Click 'Learn more' to view detailed help information with ‘Why is this important?’ for the specific selected issue
* Show the synchronized set of highlighted issues in the different panels
* **Note**: mouse and keyboard selections by the user will affect the item with current focus

**3.4 Create a scan report**

To generate a report for a single scan in the Checker view:

Open the 'Reports' dropdown menu and select 'Download current scan'.

See [Checker reports](chrome-extension://lkcagbfjnkomcinoddgooolagloogehp/usingAC.html#the_report) for more details.

**3.5 Create a multi-scan report**

To combine up to 50 multiple scans into a single report:

1. Open the 'Reports' dropdown menu and select 'Start storing scans.'

2. Scan additional pages by loading new pages and changing dynamic content with end-user input or scripting. Remember to press the Scan button on each desired state of the page(s).

3. The number of stored scans will show below Scan button.

4. Open the 'Reports' dropdown menu and select 'View stored scans'.

5. Manage the stored scans via the 'Stored scans' panel:

* **Detail**: incudes a screenshot of the page when the scan was preformed
* **Scan label**: edit the label to uniquely differentiate the scan and customize the report
* **Delete**: select one or more scans to be deleted from the report
* **All**: select all the scans by using the checkbox in the first column of the header row
* **Download**: select one or more scans to download the multi-scan report
* **Cancel**: to uncheck the selections
* **Return**: click back (or navigate) into the Checker to continue using the functionality, such as storing more scans

**3.6 Focus view**

The focus view allows you to switch between viewing all issues on the page, or only the issues for a selected element or component in the DOM.

1. Select an element in the DOM, or use the 'Inspect element' command on the page in the browser
2. Select element name in focus view (e.g., <div>) to view the related issues
3. Select 'All' in the focus view to see all issues again

**3.7 Filter views**

The Checker includes four (4) filters when viewing issues. Each filter type has a checkbox in the dropdown menu. If the checkbox is checked the issue type will show. If the checkbox is not checked the issue type will be filtered and not show. The default is for three (3) types of issues: Violations, Needs review, and Recommendations, to be checked so that they will show.

Unchecking one of the checkboxes will filter (not show) the associated issues from the list, but those issues are still counted in the total issues found and Scan summary report.

All the filters are independent. Checking or unchecking one filter does not affect the other filters. For example, if the Violations filter is checked the Violation issues will show. Unchecking the Violations filter will result in the Violations issues not showing.

**Note**: Checked or unchecked types in the Filter dropdown menu are reflected in the checkboxes in the tiles of the Scan summary and vice versa. Thus, the checkboxes in the Filter dropdown and the Scan summary behave the same and can be used interchangeably. Select or deselect checkboxes in the Filter dropdown or Scan summary to filter issues by type.

'Filter views' differs from 'Focus view' because it applies only to issue types, while 'Focus view' applies only to the selected element.

**3.8 Show/Hide issues**

The Hide feature allows issues to be removed from view and subtracted from the respective issue counts and Scan summary report. This can be useful for showing a set of issues or reflecting progress by hiding issues that have been resolved or identified to be fixed later.

To hide one or more issues, check the checkbox at the beginning of the row of the issue. Once issues have been checked the blue 'Hide' button appears in the toolbar. Pressing the 'Hide' button will hide those issues that have been selected and the counts will be updated.

**Note**: A 'Hidden' icon is displayed next to the 'Type' icon in the list of individual issues in each row, the count of hidden issues is updated, and the count of hidden issues is listed in the Scan summary. Issue counts will also be reduced for the Violations, Needs review, and Recommendations accounting for those types of issues that are now hidden.

When the 'Hidden' filter is not selected (unchecked) in the dropdown menu, then those hidden issues are filtered (not shown) from the list of issues displayed. Unlike using the Filter views feature (which only affects whether an issue is shown), Hide removes the issues from the counts and Scan summary report so they can be ignored or marked as resolved.

**Note**: When the 'Hidden' filter is selected (checked), the hidden issues reappear so they can be selected individually or as a group. When one or more hidden issues are selected, then the blue 'Show' button appears in the toolbar in the heading row. When the 'Show' button is pressed, the once-hidden issues are now shown and the counts and Scan summary report are updated.

**Resolving Needs review issues**

A common scenario is to resolve 'Needs review' issues. Follow these simple steps:

1. Check the 'Needs review' filter only (uncheck the other filter options)
2. Determine if those issues have been reviewed and confirm they are not violations and can be ignored (resolved) from further investigation
3. Confirm that those 'Needs review' type issues that have not been reviewed are not checked
4. Press the 'Hide' button to remove the checked issues

**Show previously hidden issues**

Another common scenario is to un-hide issues that were previously hidden so that they can be worked on and be included in the reports. Follow these simple steps:

1. Check the 'Hidden' filter only (uncheck the other filter options)
2. Determine if those previously hidden issues now need to be worked on and be included in the reports
3. Confirm that those ready to be worked on (to be unhidden) are checked
4. Press the 'Show' button to unhide the checked issues

**3.9 Keyboard checker mode**

This mode shows a visualization of the keyboard tab order detected on the page, and elements with detectable keyboard access issues. Use this for manual keyboard accessibility testing.

Press the 'Keyboard checker mode' icon to turn on/off keyboard visualization.

**Note**: The keyboard checker mode does not track page changes. Turn the mode off and on again to update the visualization. Remember to re-scan the page when the dynamic content has changed.

Select these icons or tab through the page to see keyboard access issues:

 tab stops numbered by tab order of the page

 keyboard access issue with tab stop number

 element not in tab stop chain with issues

**Manual keyboard testing**

Automated tools can’t find all keyboard access issues. Using the visualization, test for basic keyboard navigation:

* Make sure every interactive element is a tab stop\*
  1. No tab stops on non-interactive or non-visible elements\*\*
  2. Tab order matches the visual flow of the page
  3. Inspect elements with found issues
* Test interactive elements
  1. Start from browser URL bar, or click just above the part of page of interest
  2. Press tab key to move keyboard focus through interactive elements
  3. You shouldn’t get stuck in a keyboard trap (try the Esc key)
  4. Operate the controls using standard keys
  5. Focus should not jump to an unexpected place when operating controls
  6. When dismissing or deleting items, focus should move to sensible location

\* It may be acceptable to skip interactive elements if the UI provides another keyboard accessible way to perform the same function

\*\* It may be acceptable if the first tab stop of the page is “skip to main content” link that is not visible until it has keyboard focus

**Keys to use for testing**

* moves to next interactive element
* + moves to previous interactive element
* activates a link, button, or menu
* selects or unselects a form element, or activates button/dropdown menu
* moves between radio buttons or menu items, and scrolls page vertically
* may move between tab or menu items, scroll horizontally, and adjust sliders
* closes modal dialog or dropdown menu

Keyboard assignment conventions are defined for widgets and functions in the [ARIA Authoring Practices Guide - Developing a Keyboard Interface](https://www.w3.org/TR/wai-aria-practices/).

The Keyboard Checker Mode helps complete the critical manual testing step in the Verify phase. However, complete all the other [steps in the Verify phase](https://www.ibm.com/able/toolkit/verify) in the IBM Equal Access Toolkit to confirm the accessibility.

**4. The Assessment view**

Auditors use the ‘Accessibility Assessment’ tab because it provides an executive summary with totals for each type of issue group. It has the same functionality as the ‘Accessibility Checker’ tab but has more area to display the results because it is integrate with the ‘Elements’ panel (DOM code view) displayed by DevTools.

To launch the Assessment view, do one of the following:

* Launch Developer Tools:
  1. **Command+Option+I** on MacOS® or **Control+Shift+I** on Microsoft Windows®
  2. Or, Right-click the web page, select '**Inspect**'
  3. Select 'Accessibility Assessment' tab
* From the Chrome and Edge browser menu:
  1. Select 'More tools', then 'Developer Tools'
  2. Select 'Accessibility Assessment' tab
* From the Firefox browser menu:
  1. Select 'More tools', then 'Web Developer Tools'
  2. Select 'Accessibility Assessment' tab

After pressing the 'Scan' button, the left panel shows a detailed list of issues grouped by 'Element roles'. The right panel shows the 'Scan summary' report with totals for each type of issue group.

The default view is by 'Element roles' that shows a detailed list of issues with expand/collapse control for each group. The view is also available by 'Requirements' or by 'Rules' when selected from the dropdown menu. Auditors often organize the view by ‘Requirement’ to get a list of issues by the WCAG accessibility guideline.

Select an issue and click on the 'Learn more' link to see the detailed help information shown in the right panel that replaces the Scan summary. The help information includes applicable messages specific to the issue. The help information also includes sections for 'Why is this important?', 'Element location' with code snippet, 'What to do', 'About this requirement', and 'Who does this affect?'.

Select the number of 'issues found' above the list of issues in the left panel to return the Scan summary to the right panel.

**5. Settings**

By default, the Checker uses the latest deployment of a set of rules that correspond to the WCAG standard, plus additional IBM requirements. Use the 'Settings' page to change the default rule set for a supported guideline (standard) and the 'Keyboard checker mode' settings.

There are two methods to open the Settings page, the browser toolbar and the settings gear icon:

1. In the browser toolbar, select the IBM Equal Access Accessibility Checker icon as shown . This will usually be located in the upper right of the browser window.
2. At the bottom of the overlay that appears, select 'Settings' and the settings page will open in a new browser tab.
3. **Note**: In Firefox, the Settings page may fail to open if the Enhanced Tracking Protection option is set to Strict. To avoid this, change the browser privacy settings to Standard.

In the Checker panel itself, select the Settings gear icon to open the settings page.

**Rule set settings**

Rule sets are a packaged set of rules that are mapped to an accessibility guideline. The rules in the sets are updated regularly and each update has a date of deployment. For details on rule changes at each deployment, see the [IBMa/equal-access Release Notes](https://github.com/IBMa/equal-access/releases) in GitHub. For details on which rules are mapped to which accessibility requirement, see the [Checker rule sets](https://www.ibm.com/able/requirements/checker-rule-sets) listed in the IBM Accessibility Requirements.

The **Latest Deployment** rule set setting is the default. For consistent testing throughout a project, choose a specific deployment date. To replicate an earlier test, choose the deployment date of the original test.

Select one of the following:

* **Latest Deployment** - the latest rule set of the selected accessibility guideline (default)
* **<date> Deployment** - replicate an earlier test by choosing the deployment date of the original test
* **Preview Rules** - experiment with a possible future rule set

The rule sets for a specific WCAG version or version of the [IBM Accessibility Requirements](https://www.ibm.com/able/requirements/requirements) are available to check against an established policy. Select one of the following options from the 'Select accessibility guidelines' dropdown:

* **IBM Accessibility 7.2**: includes checking against WCAG 2.1 plus additional IBM requirements.
* **IBM Accessibility 7.3**: includes checking against WCAG 2.2 plus additional IBM requirements. This will be the default starting Oct 1, 2024.
* **WCAG 2.2 (A, AA)**: this is the latest W3C specification with no additional IBM requirements. Content that conforms to WCAG 2.2 also conforms to WCAG 2.1 and 2.0.
* **WCAG 2.1 (A, AA)**: referenced by earlier versions of EN 301 549 and other policies, but not the latest W3C specification. Content that conforms to WCAG 2.1 also conforms to WCAG 2.0.
* **WCAG 2.0 (A, AA)**: referenced by US Section 508.

**Keyboard checker mode settings**

This section contains the settings for the visualizations and notifications:

* **Lines connecting tab stops**: by default, the 'Lines connecting tab stops' checkbox is selected. Without this setting, only the numbered tab stop icons would be visible on the page.
* **Element outlines**: by default, the bounding boxes do not display. Select the 'Element outlines' checkbox to see light bounding boxes for each interactive element in the tab order.
* **Alert notifications**: toggles on and off the panel notification that appears every time the keyboard checker mode is selected. This setting is equivalent to selecting the ‘Do not show this again’ checkbox in the notification panel.

The Keyboard tap stops visualization guide notification panel is displayed each time the mode is turned on:

* **Do not show this again**: checkbox can be selected to avoid displaying the notification panel.
* **Keyboard tab stops- Visualization guide**: notification panel can be displayed at any time by selecting the ‘More Info’ icon next to the ‘Keyboard tab stops’ heading above the list issues.

**6. Checker reports**

Single scan reports are provided in both HTML web and Microsoft Excel XLS spreadsheet formats. Multi-scan reports are available only in the Excel XLS spreadsheet format. For how to generate reports, see [Create a scan report](chrome-extension://lkcagbfjnkomcinoddgooolagloogehp/usingAC.html#t_single_scan_report) and [Create a multi-scan report](chrome-extension://lkcagbfjnkomcinoddgooolagloogehp/usingAC.html#t_multi_scan_report).

**HTML reports**

This interactive report is in an HTML web format that includes:

1. Scan date and time
2. URL scanned
3. Percentage of elements with no detected violations or items to review
4. Summary of test results
5. Issue details organized by Requirements or by Rules
6. ‘Learn more’ link with detailed help description for each issue

**Note**: The percentages reported are based on automated tests only. Be sure to follow all the [steps in the Verify phase](https://www.ibm.com/able/toolkit/verify) in the IBM Equal Access Toolkit.

**Excel XLS spreadsheet report**

Both single scans and multiple scans generate a five sheet spreadsheet report.

1. **Overview** includes the name of the tool with its version, the scan date, ruleset, guidelines and platform used for the scan, and a summary of the overall results across all included scans.
2. **Scan summary** provides an overview of the set of scans within the report.
3. **Issue Summary** provides an overview of the issues found across all the scans. Issues are summarized in a prioritized order, starting with Level 1 items, as defined in the IBM Equal Access Toolkit, followed by Level 2 and Levels 3 and 4. Levels 1-3 are necessary to complete the IBM accessibility requirements. Within each level, the summary lists issues that are Violations, items that Need Review, Recommendations, and Hidden. Totals and counts are provided for each type of issue.
4. **Issues** has the details of the individual issues. This includes the scan label assigned to the scan, an ID for each issue, relevant accessibility requirements, and toolkit levels.
5. **Definition of fields** defines the columns in the other sheets.

**Note**: If the same page is scanned multiple times, there will be duplicate issues in the multi-scan report, which can be identified as having the **same** Issue ID. If a template or reused component has issues, these will also be repeated in the report, but will have **different** Issue IDs.

**7. Accessibility features**

Highlighted below are several accessibility features for improving the experience for people with disabilities using the Checker functionality, including with a keyboard or with a screen reader.

The Checker is responsive to users’ preferred font size and colors.

The Checker is fully keyboard accessible, tab order is as follows:

1. ‘Scan’ button
2. ‘Reports’ dropdown menu
3. ‘Focus view’,  toggles the switcher
4. Issue filter checkboxes,  selects checkbox
5. Keyboard Checker Mode,  toggles the mode
6. Counts for the Violations, Needs review, Recommendations, and Hidden issues
7. Total issues found,  displays the Scan summary replacing the detail help information.
8. Group issues by Element role, Requirement, or by Rule,  opens/closes the menu dropdown
9. Filter,  opens/closes the menu dropdown
10. Export scan,  downloads reports
11. Issues,  selects all issues
12. Issue groupings,  expand/collapse group
13. 'Learn more' link or the next issue
14. Individual issue,  navigate list of issues

Use the heading hierarchy and landmarks to navigate:

* **Issue count** region by issue type and the total number of issues found
* **Issue list** region with issues grouped by element roles/requirements/rules
* **Main** landmark also contains issue help and overview of stored scans
* **Scan Summary** aside or the complementary landmark contains the Scan summary report
* **Issue Help** aside or the complementary landmark contains the issue help when an issue is selected

**8. Feedback**

Visit the open-source [IBMa/equal-access](https://github.com/IBMa/equal-access/issues) GitHub repository to:

* Request a new feature
* Report a bug with the Checker
* Report a bug with a rule, help information, or the accuracy of the violation reported
* Find information on existing bugs, [Release Notes](https://github.com/IBMa/equal-access/releases), and [ReadMe’s](https://github.com/IBMa/equal-access/blob/master/README.md)

**9. Troubleshooting**

If the Checker appears unresponsive:

1. Close the browser DevTools
2. Clear browser cookies
3. Refresh the page
4. Reopen the browser DevTools
5. Press the 'Scan' button

**Note**: On rare occasions, the Checker does not appear in DevTools for some sites due to a bug in the DevTools. The workaround is to go to a site where the Checker will launch, and then launch the Checker in DevTools. Then, in the same browser tab, load the site that did not launch.

**Helpful hints**

For Chrome on MacOS, move between the keyboard tab stop visualization on the webpage and DevTools with the following keyboard shortcuts:

* When in a webpage, **Option+Command+Up** four times to get to DevTools (More or less depending on which toolbars and panels are open)
* When in DevTools, **Option\_Command\_Down** approximately five times to get to the webpage (More or less depending on the DevTools layout)

**Known issues**

* The Checker is unable to check the content of an iframe element unless both have the same origin. Work around: open the iframe URL in a new window or tab and then scan the content.
* In rare situations, the mouse pointer is unable to select underlying items on the web page when Keyboard Checker Mode tab is on.
* For certain websites on Firefox, the keyboard tab stops visualization may stay visible after turning it off and either partially work, or not work at all.
* See 'Open' issues in the open-source [IBMa/equal-access](https://github.com/IBMa/equal-access/issues) GitHub repository.
* For carousel elements, each item may be a tab stop. While only some of the carousel items are visible, the visualization will show stacked tab stop indicators for all carousel items (see image below):