Conformance and Accessibility Report

NetApp StorageGRID

NetApp
1395 Crossman Ave
Sunnyvale, CA 94089

Table of Contents

Executive Summary

Introduction

First, we would like to commend NetApp on its commitment and continued efforts to proactively provide accessible products and services. To help NetApp achieve its accessibility goals, Accessibility Partners, LLC was hired to perform a diagnostic review of the StorageGRID application.

Accessibility Partners reviewed and measured the application for accessibility conformance to the standards as set forth by Section 508 of the Rehabilitation Act of 1973, as amended. The January 18, 2018, refresh of Section 508 includes the adoption of the Web Content Accessibility Guidelines (WCAG) 2.0 for web applications. As a result, this report makes reference to WCAG 2.0 levels A and AA.

To perform an accessibility analysis, Accessibility Partners used the following assistive technologies (AT) to review individual screens, test user interface controls, and determine compatibility:
- NVDA Screen Reader
- ZoomText Screen Magnifier
- Dragon NaturallySpeaking Speech Recognition
- aXe Accessibility Toolbar
- Windows Accessibility Features

Additionally, the app was reviewed and validated for color blindness and keyboard accessibility. The compilation of these tools, along with standard accessibility test analysis, resulted in an assessment and set of recommendations, as detailed in later sections of this report.

Upon completion of the evaluation, this report has been provided for your review. This report provides specific WCAG 2.0 deficiencies, and recommendations for remediation. NetApp can use the report to review, discuss and demonstrate remediation recommendations. In addition, this report includes accessibility issues that, while not specifically WCAG deviations, represent significant issues for users of assistive technology.

Results

Based on our evaluation of StorageGRID against the Section 508 Standards, including the WCAG 2.0 Checkpoint Guidelines, we recommend improvements in the following areas (please refer to Appendix A for specific descriptions of the WCAG 2.0 Checkpoints).
- Non-text Content
- Info and Relationships
- Column Headers for Data Tables
- Use of Color

- ◆ Contrast
- ◆ Keyboard
- ◆ Timing Adjustable
- ◆ Bypass Blocks
- ◆ Page Titles
- ◆ Language of Pages
- ◆ Labels or Instructions
- ◆ Name, Role, Value

The application is currently not W3C WCAG 2.0 AA conformant; the application is also not Section 508 conformant. Note that W3C WCAG 2.0 has several levels. Section 508 conformance requires meeting level AA; this is also industry best practice for the United states and other countries.

On the positive side, our examination of the application showed that many of the problematic issues were redundant and prevalent throughout. This means that by fixing the issue once, the solution can then be applied throughout. Additionally, we believe that most, if not all of the issues diagnosed in this report can be fixed relatively easily without disrupting the integrity and purpose of the site.

WCAG Checkpoint Evaluation Principles

Principle 1: Perceivable

Principle 2: Operable

Principle 3: Understandable

Principle 4: Robust

Suggested Next Steps

It is recommended that after remediation efforts have been made, another accessibility review be conducted to check for conformance and that all items have been corrected.

StorageGRID Accessibility Detailed Report

Non-text Content

Corresponds to WCAG 2.0 1.1.1 (Level A) Guideline in the Section 508 Refresh

Images need to have alternative text that describes their purpose or meaning. If an image is for decoration only and/or used for formatting, such as a spacer it should be assigned a null (empty) ALT attribute. This tells screen readers that the image is decorative and not important for the user to know about it. Otherwise, a screen reader will announce the presence of the image and read its descriptive alt attribute when it has one.

Example of a descriptive ALT attribute for an image:

```
<img src="status_ok.gif" alt="Acceptable" />
```

Example of a decorative and/or formatting image with a null ALT attribute assigned:

```
<img src="blank.gif" alt="" />
```

The following is a sampling of areas where ALT attributes are needed in the application.

Charts – Dashboard Summary

The Dashboard summary charts are given as images with NULL alt text attributes. The information in these charts is not available elsewhere so either these charts need to have an alt text that give the info from the chart or an alternative view should be available such as a table that also displays the info given.
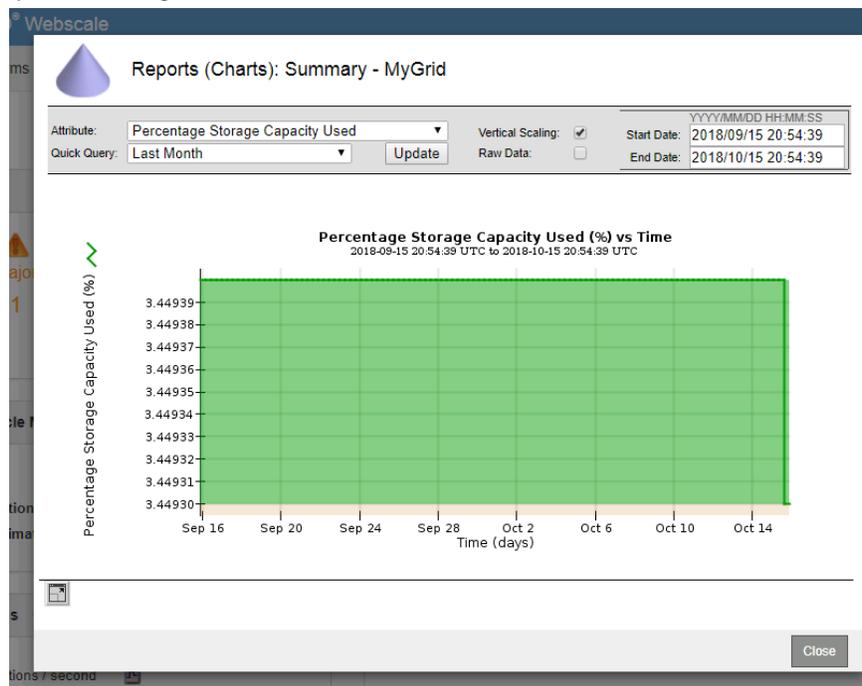
Charts – Nodes

The charts on the Nodes screen use <canvas> elements and have the issue that screen reader users are not able to retrieve the information from the charts. An additional method should be used to convey the information in the charts. See recommendations below for additional information on <canvas> accessibility.
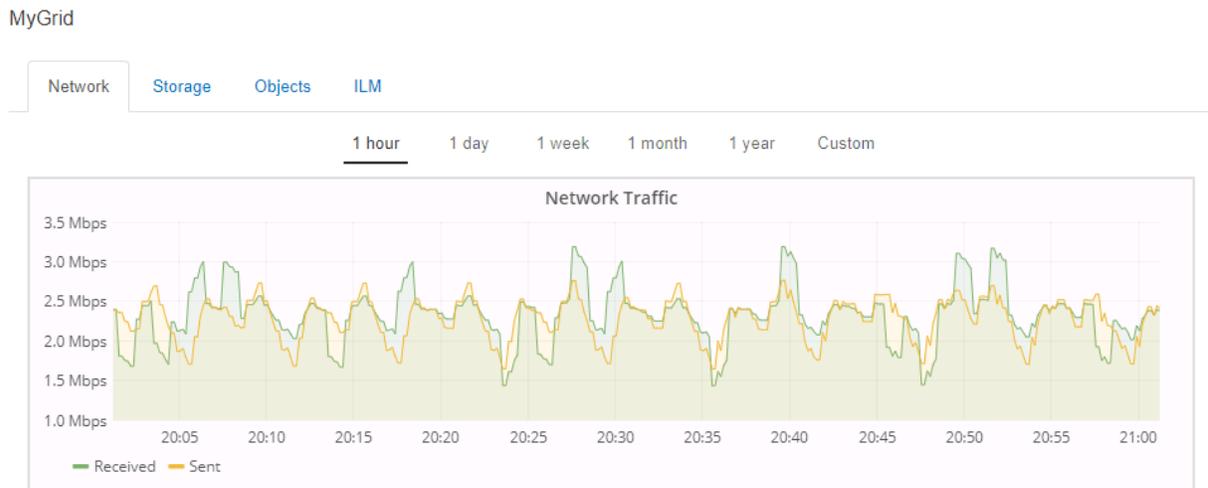
Summary Graphical Links – Dashboard

The graphical summary buttons on the Dashboard are not properly labeled for accessibility. These buttons use background images and therefore screen readers will not be able to get information from these images. These links should use <img> tags and use an ALT attribute or text should be used in the link.

Note: There are other graphical links throughout the application that have this same issue including the Add and Remove buttons when configuring Domain Names and the Information tooltips.

Table Background Images – Maintenance Decommission

On the decommission page there is a table that uses background images to display the Health of each of the grid nodes. These background images cannot be seen by screen readers so either these should instead use <img> with ALT attributes or text should be added to the cells.

Recommendation

Important and meaningful images need to have alternative text provided for them (e.g., ALT= "Help"). For decorative and/or formatting images, use empty (null) ALT attributes, i.e., ALT="". Do not use background images when such images are informative. Instead display them on the page using the IMG tag and use appropriate alternative text.

Canvas Elements: By default, HTML <canvas> elements are inaccessible. The accessibility improvement we suggest is the use of ARIA in the <canvas> element; add role="img" and aria-label="the image alt text". For further discussion and more suggestions, refer to HTML <canvas> Accessibility. http://pauljadam.com/demos/canvas.html

Info and Relationships

Corresponds to WCAG 2.0 1.3.1 (Level A) Guideline in the Section 508 Refresh

Structural markup is to be used according to its meaning, not because of the way it appears visually. The nature of a piece of content, i.e., header, bolded text, etc., can all be indicated in this way. The `<h#></h#>` elements were designed to indicate structural emphasis of Headings.

What follows are instances of areas we have found that need attention.

App

Some pages throughout the application use headings well, such as the Dashboard page. Some pages such as Alarms and Grid Options Overview do not use headings. On the Grid Options Overview page, an <h1> should be used for the main heading and then <h2> tags should be used for the sub headings of "grid information", "CA Certificates", and "Grid Options".

Recommendation

Add headings where necessary by using the heading markup, e.g., <h1>*Heading Text*</h1>. Then add styling using CSS to achieve the desired visual appearance.

Please note that screen reader users often bypass tabbing through a page completely by using screen reader shortcuts to navigate to the next header (or provide a list of all headings). Organizing a page well, as if it were an actual document, greatly improves its usability.

Column Headers for Data Tables

Corresponds to WCAG 2.0 1.3.1 (Level A) Guideline in the Section 508 Refresh

Column and row headers need to be indicated and associated with data cells for assistive technology. When this is done, assistive technology such as a screen reader can read a table's contents to the user in a manner that allows the user to understand the information. To do this for most tables with just one level of data, all that is needed is to indicate the header cells of

rows and columns using the TH tag. For complex tables, manually linking data cells to header cells using the Headers attribute may be necessary.

What follows are instances of areas we have found that need attention.

Alarms

The tables on the Alarms page do not contain table headers. Ensure that this table uses table headers similar to the table on the Tenants tab.
Note: There are other tables throughout the application that also do not contain table headers such as the tables on Storage Grades and Link Cost Configuration.

Recommendation

Use semantic markup for column headings by using the TH tags, e.g., `<th>Heading Text</th>`.

Use of Color

Corresponds to WCAG 2.0 1.4.1 (Level A) Guideline in the Section 508 Refresh

Color must not be the sole means of conveying information to the user.

What follows are instances of areas we have found that need attention.

Button Focus – Tenants

On the Tenants page, there are several instances where buttons use color only to show visual focus. Examples of this are the Tenant Accounts options such as the "Create" and "Edit Accounts" buttons. Additionally, when creating a new Tenant Account, the "Cancel" and "Save" buttons use color only to show focus. Ensure that buttons use a method other than color to show focus. This can be done with a standard focus box or other textual change method.
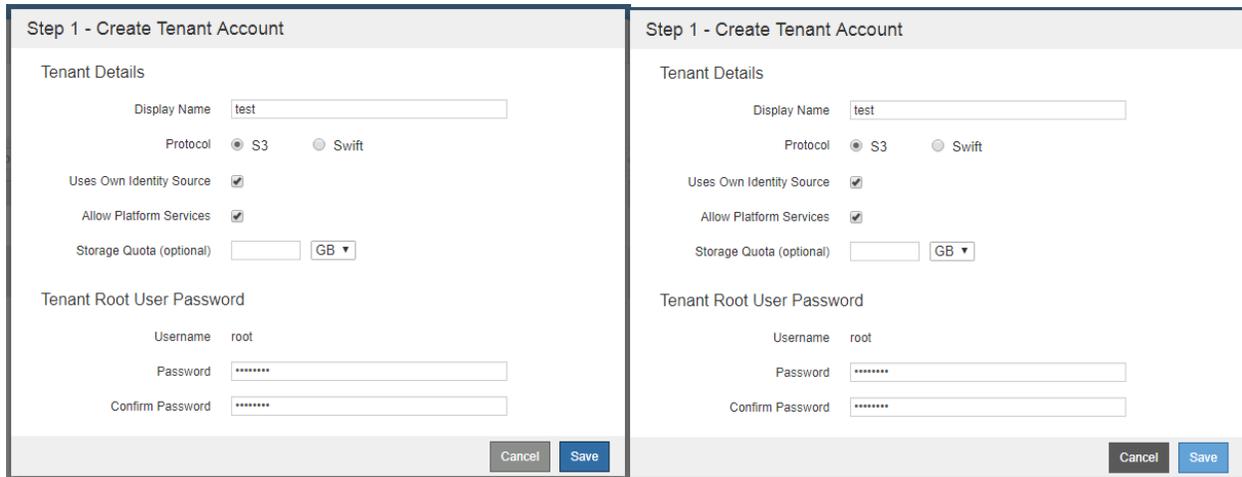
Recommendation

Using color to convey information is fine; however, color must not be the only distinguishing feature of an element. Often an asterisk or other symbol is used to provide a textual notation of color; other times, using a different shape altogether may be a more efficient method of conveying the information for all users.

Contrast (Minimum)

Corresponds to WCAG 2.0 1.4.3 (Level AA) Guideline in the Section 508 Refresh

There should be sufficient contrast between text and its background.

A minimum contrast ratio of 4.5:1 for text smaller than 18-point font or 14-point and bolded is required. For text that is at least 18-point or 14-point and bolded, a minimum contrast ratio of 3:1 is required. These ratios account for the loss in contrast due to moderately low visual acuity, congenital or acquired color deficiencies, or the loss of contrast sensitivity that typically accompanies aging.

The following is a sampling of areas and components that do not have adequate contrast ratios.

Main Menu

The main menu contains white text with a light blue background and has a contrast ratio of 2.7:1. This ratio needs to be increased according to the recommendations below in order to meet the required ratios.
Note: The light blue is used in a few other places throughout the app such as ILM policies "Simulate" and "Activate" buttons.
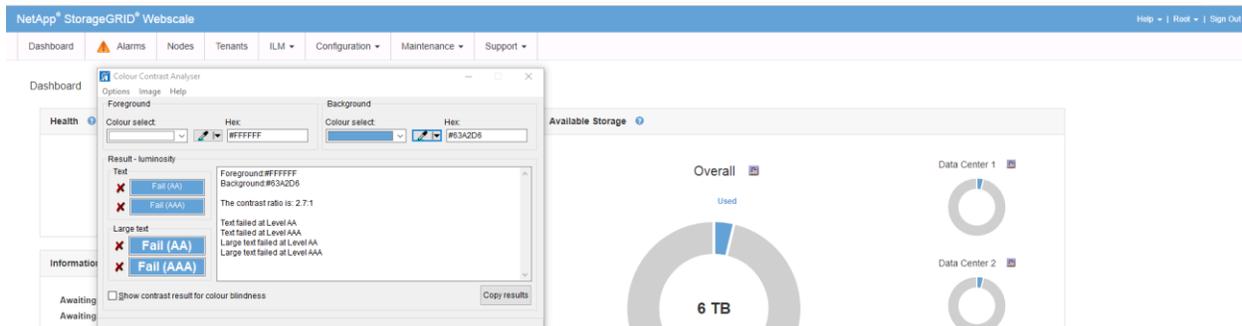
ILM Storage Grades

The table on the Storage Grades page uses some gray text with light gray background and the contrast ratio needs to be slightly increased to meet the 4.5:1 requirement for small text.
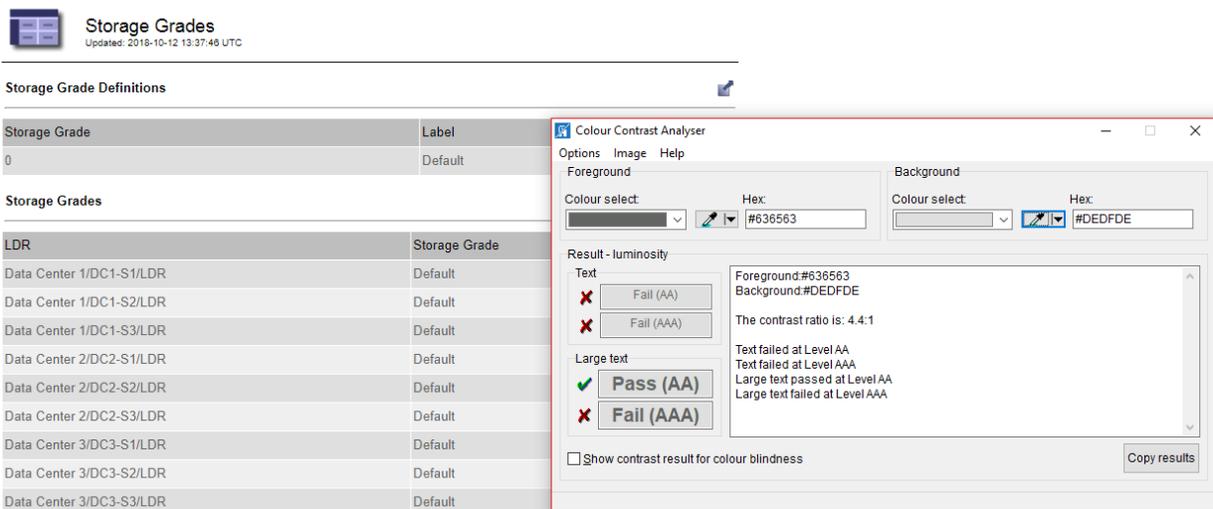
Recommendation

Increase the foreground's contrast by lightening or darkening its color to achieve a contrast ratio of 4.5:1 for text smaller than 18-point font or 14-point and bolded. For text greater than or equal to 18-point or 14-point and bolded, ensure that a contrast ratio of at least 3:1 is achieved.

Keyboard

Corresponds to WCAG 2.0 2.1.1 (Level A) Guideline in the Section 508 Refresh

The application has features and functions that are not keyboard accessible. Keyboard accessibility is required by Section 508 and WCAG, and it assists screen reader users as well as users with mobility impairments. In addition, keyboard accessibility can be a productivity boost for many users in general.

The following is a sampling of areas and components that contain keyboard accessibility issues.

Summary Links – Dashboard

The buttons on the dashboard page that opens a summary dialog are not in the keyboard focus order.
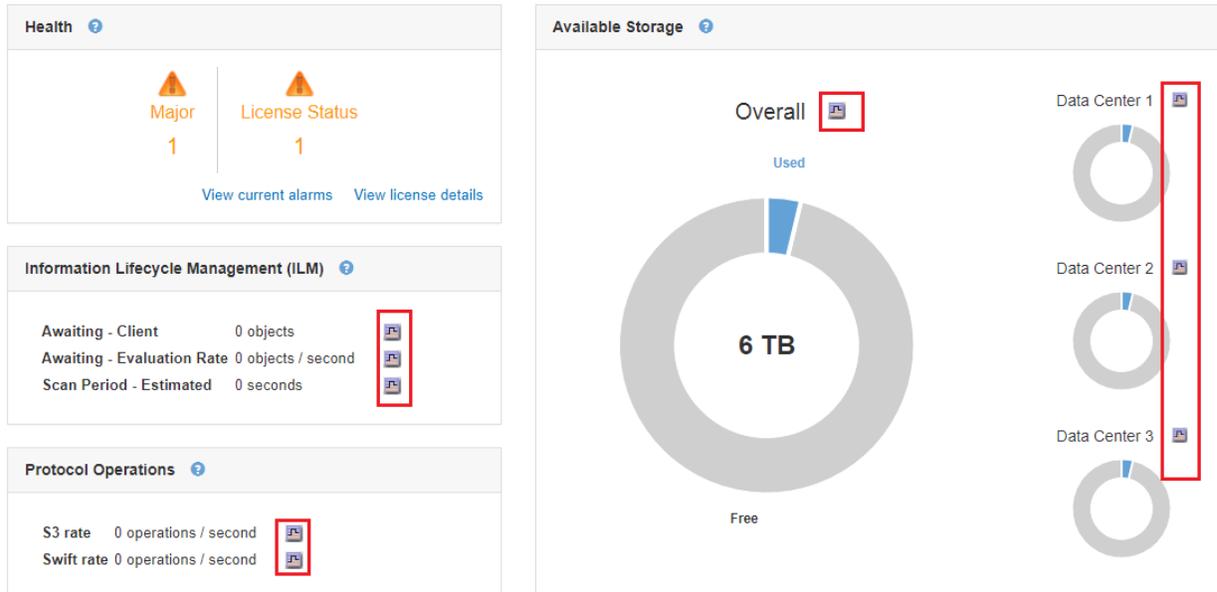
Expand/Collapse Sections – Nodes

On the Nodes page, there is a side menu that contains the grid nodes that has sections that can be expanded and collapsed. With these sections, it is not possible to expand or collapse one with the keyboard.

Timeframe – Nodes

The graph on the Nodes page has links to change the graph for a specific timeframe. These links are not keyboard operable.

Tooltips

Throughout the application there are help buttons that open tooltips for users to get help. These tooltips are not in the keyboard focus order, so keyboard users will not be able to access the help.

Table Controls – ILM Storage Pools

The table controls for inserting and removing a pool are not keyboard operable. A keyboard user can tab to these controls, but they cannot activate them.

Note: This same issue also occurs on the Storage Grades page

Graphical Buttons – Regions

Buttons to add and remove a new region are not keyboard operable.

Table Headers – Maintenance Grid Expansion

The table on this page has sortable table headers. These headers are not in the keyboard focus order, so keyboard users will not be able to sort the columns of the table.

Treeview – Support

A keyboard user is not able to expand and collapse the tree menu on the support page.
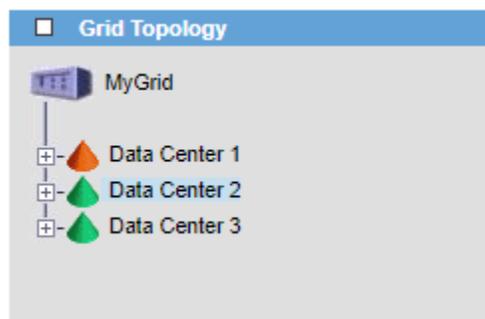
Recommendation

Ensure that keyboard access is provided. Using a standard link that calls the needed JavaScript would work. If a custom element is desired instead, then custom keyboard handling needs to be implemented. This involves setting the TabIndex property (`tabindex="0"`) on the custom element, then providing JavaScript code that monitors for when a key press such as the Enter key occurs with the element in focus.

Accessible widget libraries are also helpful when menus and tabstrips are being used. It's possible to use these directly or borrow the concepts in creating a custom one. For example, the "Accessible jQuery-UI Components Demo" (http://access.aol.com/aegis/) contains widgets such as a tabstrip and menu bar that are accessible including keyboard support. The jQuery UI has accessibility included as well. To take advantage of it, use the latest jQuery UI toolkit. See the section "Improved Accessibility" in this posting about jQuery 1.9 at: http://blog.jqueryui.com/2012/10/jquery-ui-1-9-0/. In addition, another library that offers some accessible components is the Yahoo UI toolkit at http://yuilibrary.com/.

Timing Adjustable

Corresponds to WCAG 2.0 2.2.1 (Level A) Guideline in the Section 508 Refresh

People with disabilities such as blindness, low vision, dexterity impairments, and cognitive limitations often require more time to read content or perform functions, such as filling out an online form, and often find it difficult to perform required actions before a time limit occurs, thus rendering the service inaccessible to them. Section 508 requires that websites provide flexible time limits, such as warning users before time expires, or providing the ability to adjust, extend or turn off the time limit.

What follows is an instance of an area we have found that needs attention.

App

After a period of time the application times out without warning and the user is sent back to the login screen.

Recommendation

Designing functions that are not time-dependent will help people with disabilities succeed at completing these functions. Providing options to disable time limits, customize the length of time limits, or request more time before a time limit occurs helps those users who require more time than expected to successfully complete tasks.

Bypass Blocks

Corresponds to WCAG 2.0 2.4.1 (Level A) Guideline in the Section 508 Refresh

WCAG asks that a method be provided to allow keyboard users to bypass navigational or repetitive links to get to the main content of the web page.

What follows are instances of areas we have found that need attention.

Main Menu

The main menu does not have a method to skip the repetitive navigation links.
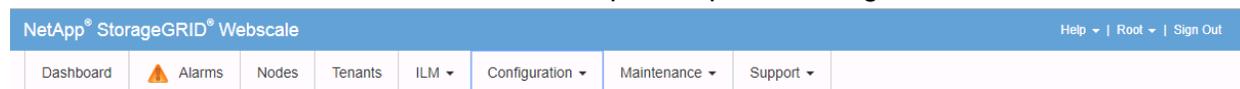
Recommendation

We recommend providing a method to bypass the repetitive links, e.g., navigational menu of links and move focus to the main area of what is being displayed. An additional method to bypass the links at the bottom of the page would also be useful to screen reader users.

The classic way to provide skip navigation is to use a same page link linked to an ID at the spot to skip to. For more information refer to the article from WebAim at http://webaim.org/techniques/skipnav.

We also recommend implementing ARIA landmarks. At present, some screen readers such as JAWS use landmarks. We expect them to play a more prominent role in navigation over time. Implementing them now is a good step in preparing the website for the future. Adding landmarks is as simple as adding "Role" attributes with the appropriate region type to the document. Most often this is added to a DIV that contains the region, such as one that contains the menu of links. There are several landmark roles; the two that will be most useful are role="navigation" and role="main."

```
<div role="navigation">Navigation Links</div>
```

For more information on the available landmark roles see http://www.w3.org/TR/wai-aria-practices-1.1/#landmarks.

Page Title

Corresponds to WCAG 2.0 2.4.2 (Level A) Guideline in the Section 508 Refresh

This criterion requires that Web pages have titles that describe their purpose. As a result, users will be able to quickly and easily identify whether the information in the Web page is relevant to their needs without having to read the page contents. Within a website, multiple pages should have unique titles to help users distinguish between them. Frames within a page should also have a title.

What follows are instances of areas we have found that need attention.

App

All pages of the app use the same title of "NetApp StorageGRID Webscale Grid Manager" and should instead convey what is on the page.

Recommendation

Ensure that each page in a web application has a `<title>` tag embedded in the `<head>` element, and that it accurately describes the page contents. Sample code is shown below.

```
<head>
  <title>NetApp Storage Grid Dashboard</title>
</head>
```

Language of Page

Corresponds to WCAG 2.0 3.1.1 (Level A) Guideline in the Section 508 Refresh

When the language of a Web page is properly set, both assistive technologies and browsers can render text more accurately. Screen readers can load the correct pronunciation rules, and visual browsers can display characters and scripts correctly. In addition, media players can show captions correctly.

In HTML, this is accomplished by using the `lang` attribute of the `html` element, as follows:

```
<!DOCTYPE html>
<html lang="en">
```

What follows are areas that we have found need attention.

App

The app does not include a lang attribute to set the language of its pages.

Recommendation

Ensure that `<html>` tags include a `lang` attribute.

Labels or Instructions

Corresponds to WCAG 2.0 3.3.2 (Level A) Guideline in the Section 508 Refresh

Labels or instructions need to be provided when content requires user input. Such form fields also need to be identified to assistive technology. This includes their Name, Role, Value and State. For example, a "Change Password" checkbox would have a Name of "Change Password", Role of "checkbox" and a State of either "checked" or "unchecked." In a web application that uses standard HTML form controls, most of this information is provided automatically including Role, State and Value. However, it is up to the developer to provide the Name (or label) information to assistive technology. If this is not done, assistive technology will make a guess to the identity of the control to try to help the user, however, it often will guess incorrectly. It is better, therefore, to remove the guesswork and ensure that the correct labels are always given to the user.

To do this, explicit labeling is used. The Label tag is used to indicate what text is to be used as the Name, which is then linked to the input field using the Label FOR attribute, as the following example illustrates.

```
<label for="username">Username: </label>
    <input type="text" name="user" id="username"/>
```

Note about the placeholder attribute: Using a placeholder has become a common practice on the web. A placeholder is not a substitute for a label and instead should be used to provide an example of what should be input.

The [W3C HTML5 specification](#) states: "The `placeholder` attribute should not be used as a replacement for a [label](#). For a longer hint or other advisory text, place the text next to the control."

The following is a sampling of areas and components that lack explicit labeling.

Combo Boxes

Throughout the web app, combo boxes are not labeled for accessibility and therefore will not give enough information to screen readers. A few examples of combo boxes that need to have proper labels are table combo boxes showing rows per page or the Storage Pool combo box on the Create Erasure Coding Profile dialog.

Checkboxes

There are many checkboxes throughout the application that are not labeled for accessibility. Ensure that all checkboxes are given a proper label, so screen readers will know the purpose of the checkboxes. Examples of unlabeled checkboxes are the Dashboard Summary Dialog checkboxes and
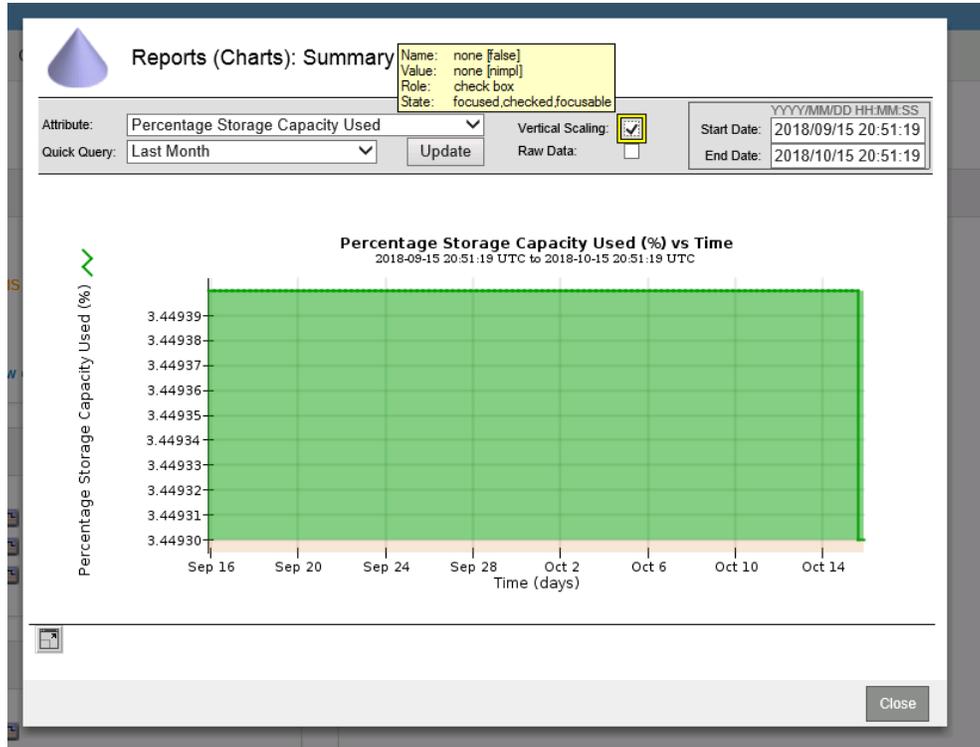
Edit Fields

Edit fields throughout the application are also not labeled for accessibility. Like the combo boxes and checkboxes above, ensure that these controls are labeled for accessibility so that screen readers can properly get the name of the field.

Recommendation

Use explicit labeling to ensure these fields are labeled for assistive technology by using the `<label>` tag and associated "FOR" attribute.

Name, Role, Value

Corresponds to WCAG 2.0 4.1.2 (Level A) Guideline in the Section 508 Refresh

Controls need to be identified to assistive technology. This includes their name, role and state. For example, a "Change Password" checkbox would have a name of "Change Password," role of "checkbox" and a state of either "checked" or "unchecked."

What follows are instances of areas we have found that need attention.

Tooltips

Tooltips throughout the application are not accessible to screen readers, these controls should use a button control that has a popup with the role of tooltip. See recommendations below for help with making an accessible tooltip.
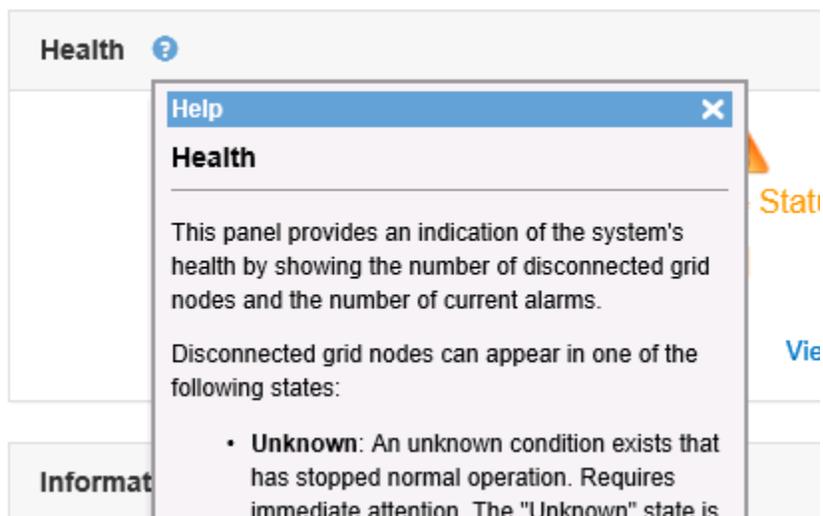
Expand/Collapse Sections – Nodes

On the Nodes page, there is a side menu that contains the grid nodes that has sections that can be expanded and collapsed. With these sections, proper information is not given to screen readers. Each section needs to announce that it is expandable, and its current state also needs to be given. See accordion recommendations below for an accessible example of an expand/collapse section.

^ MyGrid

　　^ Data Center 1
　　　　🟢 DC1-ADM1
　　　　🔺 DC1-ARC1
　　　　🟢 DC1-G1
　　　　🟢 DC1-S1
　　　　🟢 DC1-S2
　　　　🟢 DC1-S3

　　^ Data Center 2
　　　　🟢 DC2-S1
　　　　🟢 DC2-S2
　　　　🟢 DC2-S3

　　^ Data Center 3
　　　　🟢 DC3-S1
　　　　🟢 DC3-S2
　　　　🟢 DC3-S3

Tabstrip – Nodes

The tabs on the Nodes page and elsewhere such as the Support page do not give proper information to a screen reader. The tabs should have a role of tab and should also give their current state of selected or unselected. See tabstrip in recommendations below for further help with tabstrips.

```
Name:   "Storage"
Value:  "https://dc1-adm1.demo.netapp.com/grid/"
Role:   link
State:  focused,focusable,linked,traversed
```

Network　　Storage　　Objects　　ILM

Graphical Buttons – Regions

Buttons to add and remove a new region are not given a proper role. Screen readers will not know these are an actionable control and should use a button or a link instead of a <span>

Region 1　　us-east-1 (required)

Region 2　　[　　　　　　　]　　✚ ✖

Save
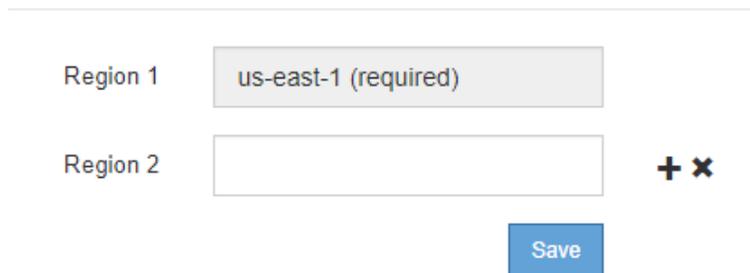
Treeview – Support

The Support page contains a treeview that allows for the user to navigate the Grid. This treeview does not give enough information to screen readers and the nodes need to announce that they can be expanded/collapsed as well as give the current level of the tree the user is on. See recommendations below for help with making an accessible treeview.
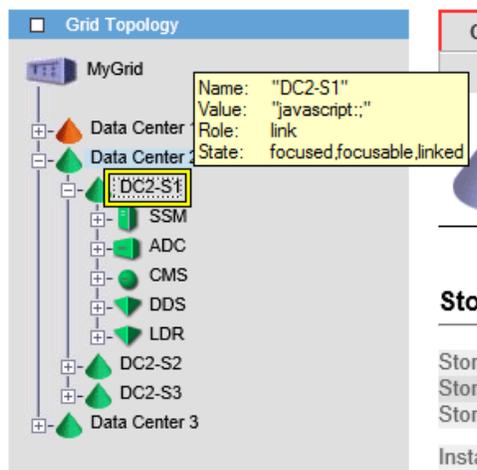
Recommendation

Tooltip:
The ARIA role of "tooltip" should be placed on the tooltip container, and the ARIA-DESCRIBEDBY attribute is placed on the input element.

See the following link for a tooltip example on an input field.
(http://accessibility.athena-ict.com/aria/examples/tooltip.shtml).

The following page explains techniques for making an accessible tooltip icon.
(https://www.sarasoueidan.com/blog/accessible-tooltips/)

Treeview: Here is an example of an accessible treeview.
(http://accessibleculture.org/articles/2013/02/not-so-simple-aria-tree-views-and-screen-readers/)

Tabstrip: Read this article on accessible tabstrips to see what can be done to make a tabstrip accessible.
(https://www.marcozehe.de/2013/02/02/advanced-aria-tip-1-tabs-in-web-apps/)

Accordion: The following article shows a method of creating an accessible accordion.
(https://www.marcozehe.de/2016/10/05/firefox-49-supports-the-html5-details-and-summary-elements/)

DatePicker:
(http://whatsock.com/tsg/Coding%20Arena/ARIA%20Date%20Pickers/ARIA%20Date%20Picker%20(Basic)/demo.htm)

Bootstrap: We note that Bootstrap is used in creating the UI. There is a Bootstrap Accessibility Plugin that may be added to a Bootstrap application to enhance accessibility. It is recommended that the latest version of Bootstrap is used for the best accessibility. See a demo and read more about its usage at the following links:

- http://paypal.github.io/bootstrap-accessibility-plugin/demo.html
- https://www.paypal-engineering.com/2014/01/28/bootstrap-accessibility-plugin-making-the-popular-web-development-framework-better/

AngularJS:
We note that AngularJS is used in creating the UI. There is an accessible module, ngAria, that may be added to an AngularJS application to enhance accessibility. Read more about its usage here: https://docs.angularjs.org/guide/accessibility. It will not solve all accessibility issues, but it's a start and will not negatively impact the UI.

ARIA:
When a custom element is created using HTML elements that were not originally designed for user interaction (e.g., a DIV element is used to create a button), assistive technology needs to know what the element represents. Web Accessibility Initiative - Accessible Rich Internet Applications (http://www.w3.org/WAI/intro/aria) (WAI ARIA) - ARIA for short - is used for this purpose.

When ARIA syntax is added to an HTML page, and a browser that supports ARIA, such as Firefox 3.0 (and higher) or Internet Explorer 8.0 (and higher), and compatible assistive technology is used, the actual intent of the custom element can be conveyed to the assistive technology, which in turn relays the information to the user. This is important because the user needs to know what something does. For example, is the element a button, in which case they would have to press Enter to activate it? Or is this list of items actually a tabstrip, in which case choosing one will update the contents of the page? These questions need to be answered in order for the user to understand the user interface and to use it efficiently. For more information, we recommend reviewing the ARIA Authoring Practices (http://www.w3.org/TR/wai-aria-practices/), as well as https://www.marcozehe.de/2014/03/27/what-is-wai-aria-what-does-it-do-for-me-and-what-not/, which provides a good overview of when ARIA should and should *not* be used.

Widget Libraries:

Accessible widget libraries are helpful when menus and tabstrips are being used. It's possible to use these directly or borrow the concepts in creating a custom one. For example, the "Accessible jQuery-UI Components Demo" (http://access.aol.com/aegis/) contains widgets such as a tabstrip and menu bar that are accessible including using ARIA roles to describe their intended purpose and keyboard support. Another library that has some accessible components is the Yahoo UI toolkit (http://yuilibrary.com/)

Parsing

Corresponds to WCAG 2.0 4.1.1 (Level A) Guideline in the Section 508 Refresh

To help assistive technologies such as screen readers accurately identify a site's content to the user, Section 508, in accordance with WCAG, requires that content can be accurately explained via its data structure and can be rendered correctly by such assistive technologies. Therefore, errors within the markup element and attribute syntax restrict accessibility; restrictions such as failure to provide properly nested start and end tags lead to errors that prevent assistive technologies from parsing the content reliably. Content, therefore needs to be parsed using the rules of the formal grammar syntax, and that all elements have complete start and end tags, are nested according to their specifications, do not contain duplicate attributes, and IDs used are unique.

HTML Validators indicate issues on all pages.

Recommendation

WCAG requires content be parsed using the rules of formal grammar syntax. To check your source code against HTML validators, use the Nu HTML Checker to get the full listings of errors for your web pages. See "Understanding SC 4.1.1 (Level A)" for additional clarification (http://www.w3.org/TR/UNDERSTANDING-WCAG20/ensure-compat-parses.html#ensure-compat-parses-intent-head).

Appendix A: Web Content Accessibility Conformance Guidelines

Principle 1: Perceivable - Information and user interface components must be presentable to users in ways they can perceive.

Guideline 1.1 Text Alternatives: Provide text alternatives for any non-text content so that it can be changed into other forms people need, such as large print, braille, speech, symbols or simpler language.

1.1.1 Non-text Content: All non-text content that is presented to the user has a text alternative that serves the equivalent purpose, except for the situations listed below. (Level A)

- Controls, Input: If non-text content is a control or accepts user input, then it has a name that describes its purpose. (Refer to Guideline 4.1 for additional requirements for controls and content that accepts user input.)
- Time-Based Media: If non-text content is time-based media, then text alternatives at least provide descriptive identification of the non-text content. (Refer to Guideline 1.2 for additional requirements for media.)
- Test: If non-text content is a test or exercise that would be invalid if presented in text, then text alternatives at least provide descriptive identification of the non-text content.
- Sensory: If non-text content is primarily intended to create a specific sensory experience, then text alternatives at least provide descriptive identification of the non-text content.
- CAPTCHA: If the purpose of non-text content is to confirm that content is being accessed by a person rather than a computer, then text alternatives that identify and describe the purpose of the non-text content are provided, and alternative forms of CAPTCHA using output modes for different types of sensory perception are provided to accommodate different disabilities.
- Decoration, Formatting, Invisible: If non-text content is pure decoration, is used only for visual formatting, or is not presented to users, then it is implemented in a way that it can be ignored by assistive technology.

Guideline 1.2 Time-based Media: Provide alternatives for time-based media.

1.2.1 Audio-only and Video-only (Prerecorded): For prerecorded audio-only and prerecorded video-only media, the following are true, except when the audio or video is a media alternative for text and is clearly labeled as such: (Level A)

- Prerecorded Audio-only: An alternative for time-based media is provided that presents equivalent information for prerecorded audio-only content.
- Prerecorded Video-only: Either an alternative for time-based media or an audio track is provided that presents equivalent information for prerecorded video-only content.

1.2.2 Captions (Prerecorded): Captions are provided for all prerecorded audio content in synchronized media, except when the media is a media alternative for text and is clearly labeled as such. (Level A)

1.2.3 Audio Description or Media Alternative (Prerecorded): An alternative for time-based media or audio description of the prerecorded video content is provided for synchronized media, except when the media is a media alternative for text and is clearly labeled as such. (Level A)

1.2.4 Captions (Live): Captions are provided for all live audio content in synchronized media. (Level AA)

1.2.5 Audio Description (Prerecorded): Audio description is provided for all prerecorded video content in synchronized media. (Level AA)

1.2.6 Sign Language (Prerecorded): Sign language interpretation is provided for all prerecorded audio content in synchronized media. (Level AAA)

1.2.7 Extended Audio Description (Prerecorded): Where pauses in foreground audio are insufficient to allow audio descriptions to convey the sense of the video, extended audio description is provided for all prerecorded video content in synchronized media. (Level AAA)

1.2.8 Media Alternative (Prerecorded): An alternative for time-based media is provided for all prerecorded synchronized media and for all prerecorded video-only media. (Level AAA)

1.2.9 Audio-only (Live): An alternative for time-based media that presents equivalent information for live audio-only content is provided. (Level AAA)

Guideline 1.3 Adaptable: Create content that can be presented in different ways (for example simpler layout) without losing information or structure.

1.3.1 Info and Relationships: Information, structure, and relationships conveyed through presentation can be programmatically determined or are available in text. (Level A)

1.3.2 Meaningful Sequence: When the sequence in which content is presented affects its meaning, a correct reading sequence can be programmatically determined. (Level A)

1.3.3 Sensory Characteristics: Instructions provided for understanding and operating content do not rely solely on sensory characteristics of components such as shape, size, visual location, orientation, or sound. (Level A)

Guideline 1.4 Distinguishable: Make it easier for users to see and hear content including separating foreground from background.

1.4.1 Use of Color: Color is not used as the only visual means of conveying information, indicating an action, prompting a response, or distinguishing a visual element. (Level A)

1.4.2 Audio Control: If any audio on a Web page plays automatically for more than 3 seconds, either a mechanism is available to pause or stop the audio, or a mechanism is available to control audio volume independently from the overall system volume level. (Level A)

1.4.3 Contrast (Minimum): The visual presentation of text and images of text has a contrast ratio of at least 4.5:1, except for the following: (Level AA)

- Large Text: Large-scale text and images of large-scale text have a contrast ratio of at least 3:1;
- Incidental: Text or images of text that are part of an inactive user interface component, that are pure decoration, that are not visible to anyone, or that are part of a picture that contains significant other visual content, have no contrast requirement.

- Logotypes: Text that is part of a logo or brand name has no minimum contrast requirement.

1.4.4 Resize text: Except for captions and images of text, text can be resized without assistive technology up to 200 percent without loss of content or functionality. (Level AA)

1.4.5 Images of Text: If the technologies being used can achieve the visual presentation, text is used to convey information rather than images of text except for the following: (Level AA)

- Customizable: The image of text can be visually customized to the user's requirements;
- Essential: A particular presentation of text is essential to the information being conveyed.

1.4.6 Contrast (Enhanced): The visual presentation of text and images of text has a contrast ratio of at least 7:1, except for the following: (Level AAA)

- Large Text: Large-scale text and images of large-scale text have a contrast ratio of at least 4.5:1;
- Incidental: Text or images of text that are part of an inactive user interface component, that are pure decoration, that are not visible to anyone, or that are part of a picture that contains significant other visual content, have no contrast requirement.
- Logotypes: Text that is part of a logo or brand name has no minimum contrast requirement.

1.4.7 Low or No Background Audio: For prerecorded audio-only content that (1) contains primarily speech in the foreground, (2) is not an audio CAPTCHA or audio logo, and (3) is not vocalization intended to be primarily musical expression such as singing or rapping, at least one of the following is true: (Level AAA)

- No Background: The audio does not contain background sounds.
- Turn Off: The background sounds can be turned off.
- 20 dB: The background sounds are at least 20 decibels lower than the foreground speech content, with the exception of occasional sounds that last for only one or two seconds.

1.4.8 Visual Presentation: For the visual presentation of blocks of text, a mechanism is available to achieve the following: (Level AAA)

1. Foreground and background colors can be selected by the user.
2. Width is no more than 80 characters or glyphs (40 if CJK).
3. Text is not justified (aligned to both the left and the right margins).
4. Line spacing (leading) is at least space-and-a-half within paragraphs, and paragraph spacing is at least 1.5 times larger than the line spacing.
5. Text can be resized without assistive technology up to 200 percent in a way that does not require the user to scroll horizontally to read a line of text on a full-screen window.

1.4.9 Images of Text (No Exception): Images of text are only used for pure decoration or where a particular presentation of text is essential to the information being conveyed. (Level AAA)

Principle 2: Operable - User interface components and navigation must be operable.

Guideline 2.1 Keyboard Accessible: Make all functionality available from a keyboard.

2.1.1 Keyboard: All functionality of the content is operable through a keyboard interface without requiring specific timings for individual keystrokes, except where the underlying function requires input that depends on the path of the user's movement and not just the endpoints. (Level A)

2.1.2 No Keyboard Trap: If keyboard focus can be moved to a component of the page using a keyboard interface, then focus can be moved away from that component using only a keyboard interface, and, if it requires more than unmodified arrow or tab keys or other standard exit methods, the user is advised of the method for moving focus away. (Level A)

2.1.3 Keyboard (No Exception): All functionality of the content is operable through a keyboard interface without requiring specific timings for individual keystrokes. (Level AAA)

Guideline 2.2 Enough Time: Provide users enough time to read and use content.

2.2.1 Timing Adjustable: For each time limit that is set by the content, at least one of the following is true: (Level A)

- Turn off: The user is allowed to turn off the time limit before encountering it; or
- Adjust: The user is allowed to adjust the time limit before encountering it over a wide range that is at least ten times the length of the default setting; or
- Extend: The user is warned before time expires and given at least 20 seconds to extend the time limit with a simple action (for example, "press the space bar"), and the user is allowed to extend the time limit at least ten times; or
- Real-time Exception: The time limit is a required part of a real-time event (for example, an auction), and no alternative to the time limit is possible; or
- Essential Exception: The time limit is essential and extending it would invalidate the activity; or
- 20 Hour Exception: The time limit is longer than 20 hours.

2.2.2 Pause, Stop, and Hide: For moving, blinking, scrolling, or auto-updating information, all of the following are true: (Level A)

- Moving, blinking, scrolling: For any moving, blinking or scrolling information that (1) starts automatically, (2) lasts more than five seconds, and (3) is presented in parallel with other content, there is a mechanism for the user to pause, stop, or hide it unless the movement, blinking, or scrolling is part of an activity where it is essential; and
- Auto-updating: For any auto-updating information that (1) starts automatically and (2) is presented in parallel with other content, there is a mechanism for the user to pause, stop, or hide it or to control the frequency of the update unless the auto-updating is part of an activity where it is essential.

2.2.3 No Timing: Timing is not an essential part of the event or activity presented by the content, except for non-interactive synchronized media and real-time events. (Level AAA)

2.2.4 Interruptions: Interruptions can be postponed or suppressed by the user, except interruptions involving an emergency. (Level AAA)

2.2.5 Re-authenticating: When an authenticated session expires, the user can continue the activity without loss of data after re-authenticating. (Level AAA)

Guideline 2.3 Seizures: Do not design content in a way that is known to cause seizures.

2.3.1 Three Flashes or Below Threshold: Web pages do not contain anything that flashes more than three times in any one second period, or the flash is below the general flash and red flash thresholds. (Level A)

2.3.2 Three Flashes: Web pages do not contain anything that flashes more than three times in any one second period. (Level AAA)

Guideline 2.4 Navigable: Provide ways to help users navigate, find content, and determine where they are.

2.4.1 Bypass Blocks: A mechanism is available to bypass blocks of content that are repeated on multiple Web pages. (Level A)

2.4.2 Page Titled: Web pages have titles that describe topic or purpose. (Level A)

2.4.3 Focus Order: If a Web page can be navigated sequentially and the navigation sequences affect meaning or operation, focusable components receive focus in an order that preserves meaning and operability. (Level A)

2.4.4 Link Purpose (In Context): The purpose of each link can be determined from the link text alone or from the link text together with its programmatically determined link context, except where the purpose of the link would be ambiguous to users in general. (Level A)

2.4.5 Multiple Ways: More than one way is available to locate a Web page within a set of Web pages except where the Web Page is the result of, or a step in, a process. (Level AA)

2.4.6 Headings and Labels: Headings and labels describe topic or purpose. (Level AA)

2.4.7 Focus Visible: Any keyboard operable user interface has a mode of operation where the keyboard focus indicator is visible. (Level AA)

2.4.8 Location: Information about the user's location within a set of Web pages is available. (Level AAA)

2.4.9 Link Purpose (Link Only): A mechanism is available to allow the purpose of each link to be identified from link text alone, except where the purpose of the link would be ambiguous to users in general. (Level AAA)

2.4.10 Section Headings: Section headings are used to organize the content. (Level AAA)

Principle 3: Understandable - Information and the operation of user interface must be understandable.

Guideline 3.1 Readable: Make text content readable and understandable.

3.1.1 Language of Page: The default human language of each Web page can be programmatically determined. (Level A)

3.1.2 Language of Parts: The human language of each passage or phrase in the content can be programmatically determined except for proper names, technical terms, words of indeterminate language, and words or phrases that have become part of the vernacular of the immediately surrounding text. (Level AA)

3.1.3 Unusual Words: A mechanism is available for identifying specific definitions of words or phrases used in an unusual or restricted way, including idioms and jargon. (Level AAA)

3.1.4 Abbreviations: A mechanism for identifying the expanded form or meaning of abbreviations is available. (Level AAA)

3.1.5 Reading Level: When text requires reading ability more advanced than the lower secondary education level after removal of proper names and titles, supplemental content, or a version that does not require reading ability more advanced than the lower secondary education level, is available. (Level AAA)

3.1.6 Pronunciation: A mechanism is available for identifying specific pronunciation of words where meaning of the words, in context, is ambiguous without knowing the pronunciation. (Level AAA)

Guideline 3.2 Predictable: Make Web pages appear and operate in predictable ways.

3.2.1 On Focus: When any component receives focus, it does not initiate a change of context. (Level A)

3.2.2 On Input: Changing the setting of any user interface component does not automatically cause a change of context unless the user has been advised of the behavior before using the component. (Level A)

3.2.3 Consistent Navigation: Navigational mechanisms that are repeated on multiple Web pages within a set of Web pages occur in the same relative order each time they are repeated, unless a change is initiated by the user. (Level AA)

3.2.4 Consistent Identification: Components that have the same functionality within a set of Web pages are identified consistently. (Level AA)

3.2.5 Change on Request: Changes of context are initiated only by user request or a mechanism is available to turn off such changes. (Level AAA)

Guideline 3.3 Input Assistance: Help users avoid and correct mistakes.

3.3.1 Error Identification: If an input error is automatically detected, the item that is in error is identified and the error is described to the user in text. (Level A)

3.3.2 Labels or Instructions: Labels or instructions are provided when content requires user input. (Level A)

3.3.3 Error Suggestion: If an input error is automatically detected and suggestions for correction are known, then the suggestions are provided to the user, unless it would jeopardize the security or purpose of the content. (Level AA)

3.3.4 Error Prevention (Legal, Financial, and Data): For Web pages that cause legal commitments or financial transactions for the user to occur, that modify or delete user-controllable data in data storage systems, or that submit user test responses, at least one of the following is true: (Level AA)

1. Reversible: Submissions are reversible.
2. Checked: Data entered by the user is checked for input errors and the user is provided an opportunity to correct them.
3. Confirmed: A mechanism is available for reviewing, confirming, and correcting information before finalizing the submission.

3.3.5 Help: Context-sensitive help is available. (Level AAA)

3.3.6 Error Prevention (All): For Web pages that require the user to submit information, at least one of the following is true: (Level AAA)

1. Reversible: Submissions are reversible.
2. Checked: Data entered by the user is checked for input errors and the user is provided an opportunity to correct them.
3. Confirmed: A mechanism is available for reviewing, confirming, and correcting information before finalizing the submission.

Principle 4: Robust - Content must be robust enough that it can be interpreted reliably by a wide variety of user agents, including assistive technologies.

Guideline 4.1 Compatible: Maximize compatibility with current and future user agents, including assistive technologies.

4.1.1 Parsing: In content implemented using markup languages, elements have complete start and end tags, elements are nested according to their specifications, elements do not contain duplicate attributes, and any IDs are unique, except where the specifications allow these features. (Level A)

4.1.2 Name, Role, Value: For all user interface components (including but not limited to: form elements, links and components generated by scripts), the name and role can be programmatically determined; states, properties, and values that can be set by the user can be programmatically set; and notification of changes to these items is available to user agents, including assistive technologies. (Level A)

Appendix B: Section 508 Criteria

Chapter 3: Functional Performance Criteria (FPC)

301 General

301.1 Scope. The requirements of Chapter 3 shall apply to information and communication technology (ICT) where required by 508 Chapter 2 (Scoping Requirements), 255 Chapter 2 (Scoping Requirements), and where otherwise referenced in any other chapter of the Revised 508 Standards or Revised 255 Guidelines.

302 Functional Performance Criteria

302.1 Without Vision. Where a visual mode of operation is provided, ICT shall provide at least one mode of operation that does not require user vision.

302.2 With Limited Vision. Where a visual mode of operation is provided, ICT shall provide at least one mode of operation that enables users to make use of limited vision.

302.3 Without Perception of Color. Where a visual mode of operation is provided, ICT shall provide at least one visual mode of operation that does not require user perception of color.

302.4 Without Hearing. Where an audible mode of operation is provided, ICT shall provide at least one mode of operation that does not require user hearing.

302.5 With Limited Hearing. Where an audible mode of operation is provided, ICT shall provide at least one mode of operation that enables users to make use of limited hearing.

302.6 Without Speech. Where speech is used for input, control, or operation, ICT shall provide at least one mode of operation that does not require user speech.

302.7 With Limited Manipulation. Where a manual mode of operation is provided, ICT shall provide at least one mode of operation that does not require fine motor control or simultaneous manual operations.

302.8 With Limited Reach and Strength. Where a manual mode of operation is provided, ICT shall provide at least one mode of operation that is operable with limited reach and limited strength.

302.9 With Limited Language, Cognitive, and Learning Abilities. ICT shall provide features making its use by individuals with limited cognitive, language, and learning abilities simpler and easier.

Chapter 5: Software

501 General

501.1 Scope. The requirements of Chapter 5 shall apply to software where required by 508 Chapter 2 (Scoping Requirements), 255 Chapter 2 (Scoping Requirements), and where

otherwise referenced in any other chapter of the Revised 508 Standards or Revised 255 Guidelines.

EXCEPTION: Where Web applications do not have access to platform accessibility services and do not include components that have access to platform accessibility services, they shall not be required to conform to 502 or 503 provided that they conform to Level A and Level AA Success Criteria and Conformance Requirements in WCAG 2.0 (incorporated by reference, see 702.10.1).

502 Interoperability with Assistive Technology

502.1 General. Software shall interoperate with assistive technology and shall conform to 502.

EXCEPTION: ICT conforming to 402 shall not be required to conform to 502.

502.2 Documented Accessibility Features. Software with platform features defined in platform documentation as accessibility features shall conform to 502.2.

502.2.1 User Control of Accessibility Features. Platform software shall provide user control over platform features that are defined in the platform documentation as accessibility features.

502.2.2 No Disruption of Accessibility Features. Software shall not disrupt platform features that are defined in the platform documentation as accessibility features.

502.3 Accessibility Services. Platform software and software tools that are provided by the platform developer shall provide a documented set of accessibility services that support applications running on the platform to interoperate with assistive technology and shall conform to 502.3. Applications that are also platforms shall expose the underlying platform accessibility services or implement other documented accessibility services.

502.3.1 Object Information. The object role, state(s), properties, boundary, name, and description shall be programmatically determinable.

502.3.2 Modification of Object Information. States and properties that can be set by the user shall be capable of being set programmatically, including through assistive technology.

502.3.3 Row, Column, and Headers. If an object is in a data table, the occupied rows and columns, and any headers associated with those rows or columns, shall be programmatically determinable.

502.3.4 Values. Any current value(s), and any set or range of allowable values associated with an object, shall be programmatically determinable.

502.3.5 Modification of Values. Values that can be set by the user shall be capable of being set programmatically, including through assistive technology.

502.3.6 Label Relationships. Any relationship that a component has as a label for another component, or of being labeled by another component, shall be programmatically determinable.

502.3.7 Hierarchical Relationships. Any hierarchical (parent-child) relationship that a component has as a container for, or being contained by, another component shall be programmatically determinable.

502.3.8 Text. The content of text objects, text attributes, and the boundary of text rendered to the screen, shall be programmatically determinable.

502.3.9 Modification of Text. Text that can be set by the user shall be capable of being set programmatically, including through assistive technology.

502.3.10 List of Actions. A list of all actions that can be executed on an object shall be programmatically determinable.

502.3.11 Actions on Objects. Applications shall allow assistive technology to programmatically execute available actions on objects.

502.3.12 Focus Cursor. Applications shall expose information and mechanisms necessary to track focus, text insertion point, and selection attributes of user interface components.

502.3.13 Modification of Focus Cursor. Focus, text insertion point, and selection attributes that can be set by the user shall be capable of being set programmatically, including through the use of assistive technology.

502.3.14 Event Notification. Notification of events relevant to user interactions, including but not limited to, changes in the component's state(s), value, name, description, or boundary, shall be available to assistive technology.

502.4 Platform Accessibility Features. Platforms and platform software shall conform to the requirements in ANSI/HFES 200.2, Human Factors Engineering of Software User Interfaces — Part 2: Accessibility (2008) (incorporated by reference, see 702.4.1) listed below:

A. Section 9.3.3 Enable sequential entry of multiple (chorded) keystrokes;

B. Section 9.3.4 Provide adjustment of delay before key acceptance;

C. Section 9.3.5 Provide adjustment of same-key double-strike acceptance;

D. Section 10.6.7 Allow users to choose visual alternative for audio output;

E. Section 10.6.8 Synchronize audio equivalents for visual events;

F. Section 10.6.9 Provide speech output services; and

G. Section 10.7.1 Display any captions provided.

503 Applications

503.1 General. Applications shall conform to 503.

503.2 User Preferences. Applications shall permit user preferences from platform settings for color, contrast, font type, font size, and focus cursor.

EXCEPTION: Applications that are designed to be isolated from their underlying platform software, including Web applications, shall not be required to conform to 503.2.

503.3 Alternative User Interfaces. Where an application provides an alternative user interface that functions as assistive technology, the application shall use platform and other industry standard accessibility services.

503.4 User Controls for Captions and Audio Description. Where ICT displays video with synchronized audio, ICT shall provide user controls for closed captions and audio descriptions conforming to 503.4.

503.4.1 Caption Controls. Where user controls are provided for volume adjustment, ICT shall provide user controls for the selection of captions at the same menu level as the user controls for volume or program selection.

503.4.2 Audio Description Controls. Where user controls are provided for program selection, ICT shall provide user controls for the selection of audio descriptions at the same menu level as the user controls for volume or program selection.

504 Authoring Tools

504.1 General. Where an application is an authoring tool, the application shall conform to 504 to the extent that information required for accessibility is supported by the destination format.

504.2 Content Creation or Editing. Authoring tools shall provide a mode of operation to create or edit content that conforms to Level A and Level AA Success Criteria and Conformance Requirements in WCAG 2.0 (incorporated by reference, see 702.10.1) for all supported features and, as applicable, to file formats supported by the authoring tool. Authoring tools shall permit authors the option of overriding information required for accessibility.

EXCEPTION: Authoring tools shall not be required to conform to 504.2 when used to directly edit plain text source code.

504.2.1 Preservation of Information Provided for Accessibility in Format Conversion. Authoring tools shall, when converting content from one format to another or saving content in multiple formats, preserve the information required for accessibility to the extent that the information is supported by the destination format.

504.2.2 PDF Export. Authoring tools capable of exporting PDF files that conform to ISO 32000-1:2008 (PDF 1.7) shall also be capable of exporting PDF files that conform to ANSI/AIIM/ISO 14289-1:2016 (PDF/UA-1) (incorporated by reference, see 702.3.1).

504.3 Prompts. Authoring tools shall provide a mode of operation that prompts authors to create content that conforms to Level A and Level AA Success Criteria and Conformance Requirements in WCAG 2.0 (incorporated by reference, see 702.10.1) for supported features and, as applicable, to file formats supported by the authoring tool.

504.4 Templates. Where templates are provided, templates allowing content creation that conforms to Level A and Level AA Success Criteria and Conformance Requirements in WCAG 2.0 (incorporated by reference, see 702.10.1) shall be provided for a range of template uses for supported features and, as applicable, to file formats supported by the authoring tool.

Chapter 6: Support Documentation and Services

601 General

601.1 Scope. The technical requirements in Chapter 6 shall apply to ICT support documentation and services where required by 508 Chapter 2 (Scoping Requirements), 255 Chapter 2 (Scoping Requirements), and where otherwise referenced in any other chapter of the Revised 508 Standards or Revised 255 Guidelines.

602 Support Documentation

602.1 General. Documentation that supports the use of ICT shall conform to 602.

602.2 Accessibility and Compatibility Features. Documentation shall list and explain how to use the accessibility and compatibility features required by Chapters 4 and 5. Documentation shall include accessibility features that are built-in and accessibility features that provide compatibility with assistive technology.

602.3 Electronic Support Documentation. Documentation in electronic format, including Web-based self-service support, shall conform to Level A and Level AA Success Criteria and Conformance Requirements in WCAG 2.0 (incorporated by reference, see 702.10.1).

602.4 Alternate Formats for Non-Electronic Support Documentation. Where support documentation is only provided in non-electronic formats, alternate formats usable by individuals with disabilities shall be provided upon request.

603 Support Services

603.1 General. ICT support services including, but not limited to, help desks, call centers, training services, and automated self-service technical support, shall conform to 603.

603.2 Information on Accessibility and Compatibility Features. ICT support services shall include information on the accessibility and compatibility features required by 602.2.

603.3 Accommodation of Communication Needs. Support services shall be provided directly to the user or through a referral to a point of contact. Such ICT support services shall accommodate the communication needs of individuals with disabilities.