

# NNELS



## Accessibility Report for Freegal

June 2022

This report was written with support from the Government of Canada's Social Development Partnerships Program - Disability Component.

The opinions and interpretations in this publication are those of the author and do not necessarily reflect those of the Government of Canada.

This work is licensed under a [Creative Commons Attribution-NonCommercial 4.0 International License](https://creativecommons.org/licenses/by-nc/4.0/).

## About NNELS

The National Network for Equitable Library Service (NNELS) is a digital public library of ebooks for Canadians with print disabilities, and an advocate for an accessible and equitable reading ecosystem for Canadians with print disabilities<sup>1</sup>. NNELS supports principles of openness, inclusion, and choice. NNELS is hosted by the BC Libraries Cooperative, a community service not-for-profit cooperative and a national leader in information and technology services.

Our team of Accessibility Testers has expert knowledge in the areas of accessibility testing, analysis, software development, and leadership. The team works to educate and advise publishers, technology vendors, and public libraries on best practices for accessibility. Our testers have lived experience with a range of

---

<sup>1</sup> Print disabilities are defined by Canada's Copyright Act and include visual, mobility, or comprehension impairments such as dyslexia.

print disabilities, including blindness, low vision, and learning disabilities.

## Introduction

Freegal is an online platform which provides access to digital audio content, including music and audiobooks. The service is available through participating public libraries as a mobile application and a website. NNELS was provided with test accounts through the Northwest Territories Public Libraries.

Our testers used screen-reading and magnification software to assess the usability of the apps and website across supported platforms. Readers can find a complete list of all the software and operating systems used in this assessment in this report's *Systems and Assistive Technology* section.

This assessment aims to determine the usability experience of readers with print disabilities and to what extent they can access content through their local public library effectively and efficiently. While this report aims to provide an overview of the accessibility performance across supported platforms, this is not an in-depth review of Freegal itself. As a result, some functionality may not be discussed at all or in-depth.

## Introduction to Assistive Technology

All mainstream operating systems include built-in screen readers (Narrator on Windows, VoiceOver on Apple devices, and TalkBack on Android) that read the contents of the screen out loud, allowing users with visual disabilities to browse apps and websites, send and receive texts and emails, and accomplish many other tasks with ease. Keyboard commands and custom touch gestures provide a flexible way for users to find and interact with the controls on-screen. Windows also has alternative screen-reading software available, most notably a commercial option called Job Access with Speech (JAWS) and a free and open-source option called Non-Visual Desktop Access (NVDA).

The text spoken by a screen reader can be sent to a refreshable Braille device. Mainstream operating systems are also equipped with user interface magnification, large text options, and high-contrast viewing modes to assist people with low vision.

To ensure the usability and accessibility of an application by people with print disabilities, all functions and controls must be accessible using assistive technologies. The DAISY Consortium explains that the basic assumption of accessibility evaluations is that reading systems "should support reading with eyes, ears, and fingers." ([DAISY Consortium, 2017](#)). It should be possible for users to read the content of the document by:

- Reading the text with screen readers or self-voicing text-to-speech (TTS) applications
- Adjusting the display, including font size, alignment, and colour contrast, or a combination of some or all these options
- Reading the text with a refreshable braille display
- Reading with assistive technologies designed for persons with dyslexia or other disabilities
- Reading with the app's built-in read-aloud functions

## **Accessibility Summary**

The Freegal apps and website present many accessibility barriers for all testers. On all tested platforms, Freegal suffers from poor heading hierarchy and page structure, inconsistent or missing button labels, and other markup issues, leading to difficulty navigating the interface with a screen reader. Low-vision testers found that the interface lacked visual adjustments and had poor or inconsistent contrast, which impacts readability and usability.

## **Accessibility Performance and Recommendations**

This section will dive deeper into specific accessibility issues encountered while testing the Freegal apps and website. Below

you will find the testing results and their related recommendations as they pertain to:

- Library Access
- Signing In
- Searching and Browsing Content
- Audio Playback

Finally, the *Development Recommendations* sections contain suggestions for improving the interface on each platform. These suggestions will be relevant to any issues or observations noted above.

## Freegal Website

### Signing In

All testers could log in successfully, but the process was unnecessarily cumbersome for screen reader users. The text fields for entering library card, PIN, and e-mail address are not labelled correctly: The text labels are readable but not associated with the text field, so a screen reader will not read them when navigating with the tab key. The close button also comes before the login button in the tab order, which is unusual enough that it may confuse some users.

### General

The navigation links for “Featured,” “Recently Played,” and “Wishlist” all have the same keyboard shortcut (alt+s) assigned to them. As a result, none of them can be reached with this shortcut.

While the “Home,” “Browse,” and “My Music” links are in a list and have the navigation landmark associated with them, these links appear after the “Recently Played” playlists section, which may cause users to miss them entirely. We recommend moving these links to the top navigation instead.

The Skip Navigation link does not appear to move keyboard focus or have any effect at all. Multiple testers were unable to use this control.

The links for accessing “Help” and “Settings” are unlabeled. Screen reader users cannot determine what these links do without activating them.

The footer of the homepage contains links to download Freegal on the Apple or Google app stores. These links are not labelled and cannot be identified by screen readers. Alt text such as “Download on the Google Play Store” and “Download on the Apple App Store” should be added.

Many pages begin with a level-2 heading. The main heading on a page should always be level-1. Screen reader users often attempt to move to the next level-1 heading as a means of quickly finding the start of the main content.

## **Item Lists**

In lists of playlists, albums, or other content, several issues exist. This applies to the search results, under the featured headings on the homepage, and when viewing a separate page for each type of item (such as by activating the “View All” link next to one of these headings.)

The cover images have alt text which is not descriptive (either completely blank or a generic label such as “Album”). Since these images are also links, many screen readers will read two links for each listed item and, using the tab key, will navigate to both the image link and the text link. This significantly increases the amount of navigation required to move through a list of items.

Since each of these list items can often have several links associated with them, navigating quickly through the list using a keyboard is cumbersome. A typical solution to this problem involves adding a heading to the name of each item and arranging the relevant information below that heading. This allows screen reader users to quickly navigate through headings

to explore the items by name without encountering additional page clutter.

In cases where these lists include buttons for each item, the buttons lack clear text labels. For example, the add to "Wish List" link has the label "Star," the download link has the label "arrow\_downward," and the menu button has no label at all. The menu button also does not have an associated state (such as "expanded" or "collapsed") which means screen readers are unable to detect whether the menu is visible or not. Finally, each menu item is lacking the "menu item" control type, which further impacts keyboard navigation.

After a "View All" link is activated, the resulting page has no <main> landmark at the start of the main content, and the top heading on the page is level 2 instead of level-1. Each sort option is also contained in a level-2 heading, making the navigation of headings confusing and verbose.

## Searches

When audio is playing, typing into the search field is often not possible because the website hijacks many keyboard shortcuts to control playback. It is not possible to press "j," "k," "l," "m," "q," the space bar, or the arrow keys (which screen reader and keyboard users must use to navigate and review text). Testers reported that this behaviour is inconsistent, with these keys occasionally being passed through to the text field despite the audio player being active.

The "Advanced Search" page has missing labels on all form fields. As is the case on other sections of the site, the text labels exist, but they are improperly associated with the corresponding controls, so screen readers cannot read them when navigating with the tab key.

Search results are displayed in a format similar to the featured items on the homepage. All issues noted in the "Item Lists" section were also observed on search results pages.

## Details Pages

After selecting a playlist, song, audiobook, or album, the items details page is shown with information about the item, shortcuts to common actions, and a tracklist (if applicable.)

The page begins with a level-2 heading, followed by an image with no alt text. The purpose of this image is unknown. The songs heading includes the "Shuffle" and "Stream" controls; these should be outside the heading.

The list of songs is visually formatted as a table, but there is no HTML table. This severely impacts screen reader navigation as there is no way to move through the table rows with a single keyboard command. Instead, a screen reader user must separately navigate through all the controls associated with a track.

Long pieces of text are truncated with a "..." in the "table" if they exceed the width of the cell. This includes song, artist, and album names. Screen reader users could not find a way of revealing the full name of the item. Since this is a solution to a visual layout problem, we recommend adding the entire name to an attribute such as [aria-label](#), which will replace the link text for screen reader users while preserving it visually.

Each song in the "table" has an unlabeled menu button. This button is also contained in an HTML list, the button being the only item in the list. This extra markup leads to screen readers announcing a lot of unnecessary information but failing to announce the important information—the purpose of the button itself. When the menu button is pressed, it does not gain an "expanded" state, so screen reader users are not given feedback about the menu being opened. It is necessary to explore further down the page to find the associated menu items. Combined with the lack of a label, this may cause users to press the button multiple times and, failing to find any additional controls, assume that the button does not work or has no relevant function. The menu items also have no control type; they should be marked up

as menu items but are identified as clickable pieces of text instead.

## Audio Player

The audio player is nearly unusable for screen readers and keyboard users due to a lack of markup on controls and a lack of keyboard navigation. The player area is also difficult to locate because it lacks a heading or landmark.

Most controls lack a label, a control type, or both. For instance, the “Share” and “Embed” controls are clickable pieces of text; the “Next” and “Previous” controls are links with the label of “Eject,” and the “Play” button could not be located at all. The “Seek” and “Volume” controls should be sliders; instead, they are likewise clickable pieces of text. The “Seek” control also has no label. Lastly, these controls are not in a logical order—especially the “Next” and “Previous” controls, which are the first thing a user encounters when navigating the player with a screen reader. The “Elapsed” and “Total” time is readable, but there is no label to identify these readouts accordingly. The volume level is not readable, and adjusting the volume is a multi-step process. An NVDA user would need to perform the following:

- Locate the “Volume” control.
- Press enter, which causes the volume to jump to approximately 50%.
- Turn on NVDA’s focus mode, which causes arrow keys and other commands to be passed through to the website rather than navigating the page.
- Press the up or down arrow keys until the desired volume is reached.

The arrow keys do not seem to adjust the volume until the keyboard focus lands on the volume control, and many users will never discover the sequence of steps required to adjust it.

## Visual Adjustment

The app contains no settings for controlling visual appearance. Low-vision testers found that the interface was generally intuitive and usable, but low contrast and inconsistent colour schemes created unnecessary barriers to readability.

## Developer Recommendations

- Use a level-1 heading at the start of the main content on every page and add headings, using the heading hierarchy to organize content.
- Label all header and footer links. This includes "Settings," "Help," and the "App Store" and "Play Store" icons.
- Hide redundant cover images on lists of songs, playlists, albums, etc. and merge the image and text links into a single control.
- Use correct markup in song/track listings. This includes using an HTML table, roles for links and buttons, and a label for the menu button.
- Add a landmark to jump to the audio player quickly.
- Ensure controls in the audio player have an associated label and role.
- Ensure all audio player keyboard commands are disabled when a text field gains focus.
- Use a simpler, more consistent colour scheme and ensure the contrast between text and background remains high.
- Ensure that all form field labels are correctly associated with their input text field so that screen readers can read them when navigating with the tab key.
- Ensure that the skip links in the website work correctly.
- Do not use the same keyboard shortcut for different menu options.

## Freegal App for iOS

- Tested iOS version: 14.6+
- Tested app version: 5.2.10

## Signing In

No issues were encountered with the login process; however, the e-mail notification screen contains a checkbox which is simply labelled "Check Box Outline Blank." This appears to be the "Do Not Show This Message Again" checkbox.

## General

Two buttons are placed near the top of the screen and are visible in all sections of the app. These buttons appear to give context-sensitive help and open the app's settings. Both buttons are unlabeled.

## Item Lists

The following information applies to many app sections: "Home," "Browse," and "Search." These sections reuse the same code to present lists of items, so the same accessibility issues are found throughout.

Lists of titles are presented to VoiceOver as single large blocks of text, causing VoiceOver to read all titles within the list at once. Since these controls represent many items at once and there is no way to simulate a tap on a specific section of text, so there is no way for VoiceOver users to choose a specific item within these lists. Each item needs to be a separate focusable control.

In lists of songs, each song has associated "Play" and "Menu" buttons. These buttons have accessibility labels of "Icon Player Play Play" and "Icon Menu Inactive More," respectively. When the menu is open, the "Close" button is labelled "Icon Close Close." Rather than fixing the labels for these buttons, the recommended approach is to hide them from VoiceOver and make use of rotor actions (see *Developer Recommendations*). Some song lists still use a single text block for the list of titles, so it can be difficult or impossible to know which song is associated with each play button.

Lastly, the category headers on the "Home" and "Browse" tabs are read by VoiceOver as static text rather than headings. This

means a VoiceOver user can't use heading navigation to jump between the various categories.

## **My Music**

The "My Music" tab does not use the same list format, so lists of items are far more usable. Testers noted that the playlists subtab presents other accessibility-related bugs: The navigation buttons in this section are incorrectly read by VoiceOver as headings. These include the buttons for switching between streaming or downloaded playlists, as well as the "Create Playlist" button.

When creating a playlist, the text fields are unlabelled, so VoiceOver users may have trouble identifying them. Once the playlist has been created, the "Menu" button still contains the incorrect label "Icon Menu Inactive More."

## **Searching**

Low-vision testers reported that the search field is quite small and can be difficult to see.

After performing a search, the top buttons are labelled "Close Button Close" and "Search Back Back." Once again, their purpose is clear, but the labels should be far more concise.

When activating the "Sort" button, the sorting options appear at the bottom of the screen without any announcement to VoiceOver. Ideally, a popup should be used so VoiceOver focus is confined to the sorting options. With the current interface, users must explore the screen to find the options, and it is still possible to read and interact with the rest of the controls.

Search results are presented in a similar format to the browse screen. All issues noted above are also present with search results: Many lists of items are presented as a single block of text, and the "Play" and "Menu" buttons are not clearly labelled.

## **Audio Player**

When audio is playing, a small player appears at the bottom of the screen. This can be expanded by pressing a button which

VoiceOver identifies as “up arrow.” Playback can be paused or resumed without expanding the player. VoiceOver identifies this button as “Icon Player Play.”

Many controls are improperly labelled in the full-screen player, though most are easy to identify. The “Back” button is labelled “Icon Arrow left,” and the “Previous,” “Play,” “Next,” and “Shuffle” buttons all contain the word “Icon” in their label. The “Elapsed” and “Remaining” timestamps are readable, but the controls would benefit from clearer labels so VoiceOver users can more easily determine which is which.

### **Visual Adjustment**

The Freegal app contains no visual adjustment settings. Low-vision testers found the interface to be more visually consistent than the website, but additional font and colour adjustments would be helpful. Additionally, some interface elements—such as the search field and the arrow to reveal the audio player—are quite small and easy to miss.

### **Developer Recommendations**

- Change button labels to be more concise and consistent, eliminating words such as “icon” and ensuring their function is clear
- Consider using rotor actions to expose the “Play” and “Menu” functions in lists of multiple items. This reduces the amount of screen reader navigation required to move from one list item to the next.
- Separate lists of multiple items into individual controls so a screen reader user can select an item within the list
- Remove redundant heading roles from buttons.
- Use appropriate popup controls to trap VoiceOver focus when displaying Sort options.
- Increase the size of small icons and controls.
- Add font and colour adjustments to the app or mirror system settings where appropriate.
- Ensure the category headers on the “Home” and “Browse” tabs are read by VoiceOver as headings.

## Freegal App for Android

- Tested Android Version: 11 and 12
- Tested app version: 5.3.7

### Signing In

Screen reader users noted that the "Back" button has no label when choosing a library. The "Need Help" control is labelled but is not identified as a button.

### General

Several controls are available in every part of the app. The top of the screen contains "Help" and "Settings" buttons, both with no text label. Icons for "Home," "Browse," "Search," and "My Music" are available as bottom tabs. Screen readers identify these buttons as icons, which is technically correct but unconventional. Since they are actionable controls, announcing them as buttons or tabs would make more sense.

Whenever a playlist, album, or audiobook is selected, a common interface will be displayed with a list of "Tracks," "Shuffle," and "Sort" buttons, and information about the length and number of tracks. Each track has an unlabeled menu button next to it, and the "Back" button is also missing a label. The "Sort" and "Shuffle" controls are missing a control type.

### Item Lists

The following information applies to many sections of the app, including "Home," "Browse," and "Search." These sections reuse the same code to present lists of items, so the same accessibility issues are found throughout.

In many areas of the app, lists of items will be presented with various categories separating them. For instance, in the "Featured" section of the "Home" tab, the items will be grouped under headings such as "Recently Played" and "Featured Playlists." TalkBack does not identify these section headers as headings, and the "View All" control is not identified as a button.

Adding the heading role to these category headers will allow TalkBack users to jump quickly between them.

Many types of content allow the user to open a menu with available actions, which can differ depending on the content type. The menu items are accessible; however, the menu buttons next to each list item are unlabelled, as is the button for closing the current menu.

In lists of genres on the "Browse" and "Search" tabs, each list item is reported to screen readers as non-clickable, causing TalkBack to make a different sound and avoid announcing the typical "double-tap to activate" hint.

The "Playlists" section in the "My Music" tab has additional controls for managing playlists. The "Create Playlist" button is missing its control type, along with all buttons within the "Create Playlist" dialogue.

## **Audio Player**

The audio player becomes visible when a track has been selected. Initially, it appears as a small bar at the bottom of the screen with controls for opening the full-screen player and pausing or resuming playback.

In the full-screen player, the "Play/Pause" button is missing both a label and control type. The elapsed and remaining time cannot be read with a screen reader, only the position percentage. The position slider is also not adjustable with a screen reader.

When playing a video, the player interface is completely different and not accessible to screen readers at all. Screen readers detect only one control, which is not labelled and has no noticeable effect when pressed.

## **Visual Adjustment**

The app has no visual adjustment settings. Testers found some interface elements—such as menu items—are small and difficult to see due to poor contrast.

## Developer Recommendations

- Label all “Back,” “Menu,” and “Close” buttons throughout the app and any other buttons lacking a text label.
- Add the “button” control type to actionable controls where necessary.
- Improve contrast to make interface elements easier to see.
- Remove the word “Icon” on the bottom tab labels.
- Add the heading control type to category headings.
- Fix the video play so that it is screen reader accessible
- Increase the size of small icons and controls.
- Add font and colour adjustments to the app.

## Conclusion

The Freegal apps and website present many accessibility barriers, which differ depending on the chosen platform. Screen reader users will be unable to browse most types of content on the iOS app, and all platforms suffer from poor button labels and missing control types. The web interface has an inconsistent colour scheme, often leading to poor contrast between foreground and background. Since many interface elements are reused in multiple sections of the interface, any accessibility fixes made to the apps and website would also affect more than one screen. NNELS hopes that the observations and suggestions in this report will help users and developers find and create a more accessible user experience.

## Systems and Assistive Technology

- Operating Systems
  - Windows 10 (21H1)
  - iOS 14+
  - Android 11 and 12
- Mobile Applications
  - Freegal 5.2.10 (iOS)
  - Freegal 5.3.7 (Android)
- Browsers

- Chrome 90, 91
- Screen-readers
  - NVDA 2020.4 (Windows)
  - JAWS 2021 (Windows)
  - VoiceOver Touch (iOS)
  - TalkBack (Android)
- Magnification
  - Magnifier (Windows)
  - ZoomText 2020 (Windows)
  - Zoom (iOS)

## Acknowledgements

The following testers and editors contributed to this report:

- Simon Jaeger (Lead Writer)
- David Kopman
- Riane LaPaire
- Ka Li
- Laetitia Mfamobani
- Deanna Ng
- Megan Sellmer

Published by the [National Network for Equitable Library Service](#) (NNELS), Vancouver, BC, 2022