



## Accessibility Testing of OverDrive

Published by the National Network for Equitable Library Service (NNELS), Vancouver BC, June 2020

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](#)

## About NNELS

The [National Network for Equitable Library Service \(NNELS\)](#) is a digital public library of ebooks for Canadians with print disabilities<sup>1</sup>, and an advocate for an accessible and equitable reading ecosystem for Canadians with print disabilities. 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 print disabilities, including blindness, low vision, and learning disabilities.

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

## Introduction

The Libby platform is the successor to OverDrive, and provides access to ebooks and audiobooks through local libraries.

This report will cover ebooks and audiobooks. Our testers used screen reading and magnification software to assess the usability of the Libby apps and website across supported platforms. The objective of this assessment is to determine the usability experience of readers with print disabilities<sup>2</sup>, and to what extent they can access audiobooks and ebooks through their local public library effectively and efficiently. It is important that readers with print disabilities can choose the reading systems that offer the accessibility features they require.

The results of our evaluation show that testers encountered many accessibility bugs, including unlabeled or mislabeled controls, confusing navigation, and an inability to read the text of ebooks. This report highlights the accessibility barriers of the Libby apps and website, explains the barriers that these issues pose and why they are problematic, and advances some recommendations to enhance the usability experience for readers with print disabilities.

Libby 4.0 was released for iOS and Android at the beginning of June, 2020 with “behind-the-scenes improvements to pave the way for language support and accessibility”. Our testing began before version 4.0 was released, but this document has been revised to reflect the changes found in this update. Most notably, improvements have been made to the audiobook player and ebook reader to create fully accessible navigation; however, it is not possible to read the text of an ebook with screen readers. Our hope is that this is only the beginning of a larger change to development practices that could lead to a more accessible Libby.

## 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 and accomplish many other tasks with ease. Keyboard commands and custom touch gestures provide a flexible way for a user to find and interact with the controls on-screen. Windows also has alternative screen reading software available, most notably a commercial option called JAWS (Job Access with Speech) and a free and open-source option called NVDA (Non-Visual Desktop Access). Text spoken by a screen reader can be sent to a refreshable braille

<sup>2</sup> Print disabilities are defined by Canada's Copyright Act and include visual, mobility, or comprehension impairments such as dyslexia.

device. Mainstream operating systems are also equipped with user interface magnification, large text options, and high-contrast viewing mode to assist people with low vision.

To ensure usability and accessibility of an application by people with print disabilities, all functions and controls must be accessible using assistive technologies (including screen readers, screen magnification software, and switches and voice controls for people with mobility impairments). 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) functions
- Adjusting the display including font size, alignment, and color contrast, or combination of some or all of these options
- Reading the text with a refreshable braille display
- Selecting font types designed for persons with dyslexia or use software designed for people with learning disabilities
- Reading with the app’s built-in read aloud functions

## Accessibility Performance and Recommendations

In this section we dive deeper into specific accessibility issues of the various features of the Libby platform. Below you will find the testing results and their related recommendations as they pertain to:

- Library Access
  - Signing In
  - Searches
  - Downloading Titles
- Reading and Navigating Content
- Creating Bookmarks
- Visual Adjustments

We highlight some of the problematic issues as they relate to the different parts of the application in each of the platforms tested, but we do not elaborate on each of the features.

### General

The Libby platform is powered entirely by a web interface that is rendered similarly in a web browser and in the app for each operating system. This is a development practice that has become increasingly common throughout the past decade because of the relative ease with which a developer can create a cross-platform app. This section of the report will cover usability issues which are reproducible across most, if not all, operating systems, while issues unique to specific operating systems will be covered in the following sections..

All versions of the app bring up a message for screen readers when launched. This message advises users that “we are working to improve your experience with this app. In the

meantime, our OverDrive app is more accessible.” The message does not disappear when the *hide* button is pressed; in fact, one tester observed that after swiping right on a mobile device, the top of the iOS and Android main screen is obscured by this button, which blocks off part of the screen where a menu would normally appear. On Android, a swipe left gesture makes this button go away, but not on iOS.

The screen reader message changes as the user navigates the app, reflecting the section that is currently loaded. For instance, bringing up the *Library* screen causes the top of the page to say “You are at library.” As it stands, these hints are quite helpful; however, they still block off the top part of the screen and the hide button does not seem to work (it does nothing). There is also room for improvement in the alerts that appear when opening a book; the alert simply says “You are at open.” It’s helpful to know the screen has changed, but hearing the name of the title that is opened would be ideal.

From looking at the page DOM<sup>3</sup>, we can see that these alerts use a div with the alert role. This is a valid approach for new content loading, but since a screen reader announces it automatically, it doesn’t need to be at the top of the page and it doesn’t need to stay on the screen.

Most websites and apps don’t use this approach because the top of the screen typically has a heading that denotes the current section. Libby doesn’t currently do this, which is the primary reason why these hints are helpful. Hopefully a future update to the interface will bring a more accessible design, thus making these alerts unnecessary.

No meaningful separation exists between sections of the screen. As a result, as the main screen fills with more information, a screen reader user will have increased difficulty navigating to the various sections of the app. Groups of buttons and other controls may be visually set apart from each other, but adequate use of headings, landmarks, and other meaningful controls would help to create a universally accessible experience.

Screen reader users may also face confusion when selecting any buttons or links that cause new content to temporarily load over the current screen, such as a search filter, an actions button, or a menu. While the new controls are visually obvious, a screen reader user is not told that new content is loaded or focused on the new controls, and there is no consistency to the location of new content compared to the button that activated it. This can lead to a situation wherein a user presses a button and needs to then examine the entire app from top to bottom in order to find the new controls. This is a particularly frustrating process on a non-touchscreen device where the user might be moving systematically down the page using the arrow keys, rather than exploring by touch and intuiting the location of popovers.

<sup>3</sup> The Document Object Model (DOM) is a programming API for HTML and XML documents and defines the logical structure of documents and the way a document is accessed and manipulated.

## **Signing In**

Testers on all platforms were able to sign in using their local libraries. The current sign-in interface leaves the previous steps visible to the screen reader and loads new content below them. As a result, advancing to the next step does not automatically warn the user that new content has loaded, nor does it move the cursor to that new content. There are also no headings or other meaningful separation between the steps. This can be confusing to new users and the lack of headings makes navigating to the next step unnecessarily difficult. On a touchscreen device, the user can explore the screen to find that new content has loaded; however, without a touchscreen, there is no way to tell new content has loaded other than deciding to manually scroll down through the controls of the page—something users may not do if they do not expect new content to silently load.

While Libby helpfully includes an option to log in using a code from an existing device, the screen reader user is not told which button to press on their other device in order to retrieve this code because the example button does not have a label. Screen readers will simply say “button.”

## **Searching for Books**

This applies to both ebooks and audiobooks. Testers on iOS and Android reported that the search box did not appear in a predictable location in swipe order (in other words, swiping left or right through the controls did not move the user to the search box as expected.) Users were sometimes able to find this control by swiping left after touching the bottom of the screen, and as always, it can be located by touch as long as the user knows where it is.

While the category filters are accessible, the user is not warned that additional options have appeared after opening a filter. The filter options have no relation to the links that activate them, so they are hard to find with keyboard commands or swipe gestures that move the user through controls in a systematic way.

Testers noted that it was time consuming to navigate through the search results, as they were not clearly separated. For example, whenever a book has reviews, the five images of stars require excessive navigation (5 swipes on Android and 12 on IOS) to move past them to the book’s title. One way of separating information, which is compatible with screen readers, is to add headings. By creating a clear and navigable separation of sections, users will have the ability to quickly move focus to different elements by using shortcuts to buttons, headings, landmarks, and more.

The lack of separation in popovers made filtering and refining searches particularly cumbersome. Users are not warned that additional options have appeared after the button is pressed, and the location of the filter and refine options meant that testers needed to move to the bottom of the screen in order to find the relevant controls. This was a minor inconvenience on touchscreen devices, but much more difficult with a keyboard where navigation is far less spatially oriented.

Testers had no trouble borrowing books and navigating the bookshelf. The action buttons next to titles have clear labels.

## Audiobooks

Since our testing in July of 2019, Libby 4.0 has released a new audiobook player on iOS, Android, and the Libby website, providing an accessible and pleasant experience. The time readouts are labeled, as are the buttons to play / pause, skip backward / forward, and open the chapter list. Selecting a chapter will now activate a jump directly to that chapter, and it begins playing. Bookmarks can be added, edited, deleted, and navigated to using well-labeled buttons and popovers that keep focus inside them. Libby 4.0 was released shortly before the writing of this report, and this particular part of the app appears to have had significant work done to improve it, which suggests that similar changes could come to other parts of the app in future.

The Windows app is excluded from these changes as of this report, so Windows users may wish to access the website instead.

## Ebooks

Much like the audiobook player, many changes have been made to the ebook reader across iOS, Android, and the website. VoiceOver users can now locate buttons to jump to the next or previous page, as well as a chapter list similar to the one seen in the audiobook player. Unfortunately, the text of a book cannot be read by a screen reader, so this part of the app is still not accessible.

## Visual Adjustments

All tested platforms have a dyslexic font available, which is excellent. It is not possible to change the foreground and background colour independently of a preset style. Testers noted that the font selection was different across each platform—most likely based on available system fonts, with Android having very few to choose from. Alignment can be changed, and the page view in ebooks can be adjusted between single or dual page.

## Developer Recommendations

- Ensure screen reader users can access the text of ebooks.
- Review all buttons and controls to ensure they have a text label.
- Consider moving the alerts to a different part of the page and hiding them after some time, or removing them entirely and using headings to denote the current section.
- Reorder the controls so that the order in which they are rendered matches the order in which they are presented on-screen.
- Add meaningful separation between items: nav and main Landmarks go a long way toward breaking up the app into relevant sections. Headings between search results will help the user navigate them quickly.
- Add headings between the individual steps of the sign-in process and, if possible, refocus the user on each new step once it is visible.
- Use standard controls and meaningful separation when content is temporarily added or painted over the existing interface, such as search filters, menus, or table of contents. This could be a dialog for large screens of additional information, and a menu

or picker for filters that only require a single selection. Place a user's focus in this container when it is selected.

- Ensure users are able to access settings and other important sections of the app.
- Ensure users are able to fully customize font type and size, foreground and background colours, and line spacing. There can never be too much visual customization, and a user should not need to make a choice between one customization or another.

## iOS

- Tested in iOS version 13.4
- Tested app version: 4.0

In addition to the general information above which applies to all operating systems, the following platform-specific observations were gathered by testers using iOS devices.

The swipe left and swipe right gestures should move VoiceOver's navigation systematically through the controls in an app, starting from the top-left and ending at the bottom-right. However, in Libby—particularly on the main screen—these gestures don't present the user with a logical control order, making navigation unnecessarily confusing. Occasionally, VoiceOver will also allow the user to touch or swipe to text and controls that are no longer visible. VoiceOver (and therefore the user) will assume the controls are still clickable and present on-screen, but in reality, the user may need to scroll or dismiss a popover in order to properly interact with them. We refer to this phenomenon as "backfield interference". It occurs when new controls are painted over existing ones without first clearing the screen of its old content, and the user is still able to move to the old controls.

There was also an issue with the screen reader maintaining focus for popovers. This meant that the tester had to search the entire screen to locate the popover, since its content mingles with the rest of the content on the screen. This lack of focus for screen readers when interacting with popovers makes them not only hard to find, but also hard to close. They also reported that pull-down menus cannot be closed at all, with one tester reporting that this particular issue resulted in them needing to restart the app 14 times in a 6-minute period.

## Searching for Books

This applies for ebooks and audiobooks. Testers noted that the search field is properly labeled, however, it is difficult to determine where it is located since the flick order does not match the positional elements on the screen—as mentioned in earlier in this report. Having said that, one of our testers was quickly able to locate the search field when they moved to the bottom of the screen and then swiped back a few times. Overall, there were different results from different testers, with some being able to find the search field at the bottom of the screen, while others could not find it at all. In these cases it appeared to be hidden, even as these testers explored the same area of the app.

The "Refine List" element was noted to be cumbersome and time consuming to use. Testers noted they were able to filter criteria on the list, but they had to go in and add one item at a

time, and then click on "refine refine list", as they could not choose all the options they wanted at once. They also noted that each category has the word "chevronright" at the end of each category title, such as "language 1 of 1 chevronright".

### Developer Recommendations

- To avoid the backfield interference issue described above, ensure previous content is completely cleared from the screen when new content is added.

### Android

- Tested in Android version 10
- Tested App Version: 4.0

In addition to the general information which applies to all operating systems, the following platform-specific observations were gathered by testers using Android devices.

Similar to the iOS app, TalkBack can sometimes navigate to controls that are not visually present on-screen or are covered by a popover. This backfield interference adds an extra layer of complexity to the app, as the user sometimes needs to explore the screen thoroughly just to determine which information is meant to be in focus.

### Signing In

After the testers entered their library's name, they were then presented with a button for the correct branch. When they selected this button there was no response. The tester discovered that they first had to close the keyboard before the button would appear to be clickable. There were also issues reported with the Library Card and Pin boxes. The Pin box area is directly after the Account Number, but it could not be filled until the "Next" button was selected.

### Searching for Books

This applies to ebooks and audiobooks. Testers found that they could search for titles and navigate through the list of search results. Nevertheless, they did report a similar issue to VoiceOver with the options at the bottom of the screen being hard to locate and navigate to. If a user relying on TalkBack chooses either the "List Preferences" or "Refine Refine Search," they have to explore by touch to find where the options appear on the screen. Testers noted that if they swiped to locate these options, it would be time consuming due to issues with scrolling using gestures. For instance, while users can flick to the right from the top field, when they try to swipe to the left from the bottom, the focus gets stuck at the top boundary web view element containing the content. This makes it difficult to go to an element near the bottom of the screen.

### Visual Adjustments

It is easy to change font size, colour, justification, and style. Though, again, it would be better to have more customizable colours, so all readers can create a contrast that works best for them. It is good to see that the dyslexic font is available. The tester did discover

though that the number of choices for other font types was very limited, with only Serif and Sans Serif as options. It should be noted that the iOS app has six choices.

### Developer Recommendations

- More fonts and completely customizable colours would create a better reading experience.

### Website

The website was tested on mobile and desktop operating systems, so unless otherwise specified, all information below pertains to both. Since many of the findings are the same across different browsers and operating systems, this information will be relevant to any software combination unless otherwise noted.

The experience of using the Libby website is almost identical to that of iOS and Android. Using Libby in a browser on Windows is also notably different—and better—than using the Windows app for several reasons: The book viewers are updated to use the new and accessible interface described above, although the text of ebooks is still not readable. The website also does not have blank lines scattered throughout it.

The Windows app and website both notably remove the ability for users to navigate using the tab key—something which many keyboard users might want to do. This may have been unintentional, as testers could not find any sections of the website which reuse the tab key for another purpose.

### Visual Adjustments

As is the case with all versions of Libby, colours can only be changed by selecting predefined themes. Testers did find a wide selection of fonts available for reading ebooks.

### Developer Recommendations

- Restore the ability to navigate the website using the tab key.

### Windows

- App version tested: 1.4.2

Libby's Windows app is still a web interface, but appears to have some significant differences that make it a much less accessible option than simply using a browser to access the website. Most notably, the book viewer is completely inaccessible on this platform whereas the website and mobile applications have been updated with a much more screen reader friendly interface, particularly for audiobooks.

Most screen reader users utilize desktop operating systems with a keyboard, and finding a control by guessing its location on-screen is not possible; screen reader users rely on the ability to move between links, headings, buttons, text fields, and other controls using keyboard commands. In most apps, the tab or shift+tab keystrokes will move forward and

backward through controls. If none of these commands function well, it is still possible to move through a screen reader's "virtual rendering" of an interface using the arrow keys.

For reasons unknown, the tab key does not navigate this app properly. Instead, it cycles between the main screen and a blank page. This poses a significant barrier to screen reader users and others who cannot utilize a mouse for navigating the app.

Libby's lack of separation between sections of the page and control groups has been discussed previously in this report: The main content and top and bottom sections are not given any kind of meaningful element surrounding them, such as a nav and a Main landmark, or a heading. As a result, navigating the interface on Windows is often confusing and time-consuming because there is no easy way to move between major sections of the page other than using the arrow keys. On a touchscreen device, this can instead be worked around by remembering the location of controls and touching them directly.

Overall, the screens were found to be cluttered with too much information and no options for navigation. For instance, a tester noted that after opening the "Help" and "Advanced Search" features, they could not close them. As these features stayed open, they added further clutter to the interface. Therefore, the longer they used the app the harder it became to use. The only way to declutter the screen was to close and reopen the app.

A consistent approach is crucial to efficient navigation with adaptive technology, and very little seemed to be consistent in this app. For instance, at some point throughout one tester's exploration of the app, the buttons to return to Library and Bookshelf vanished from the bottom of the page.

The lack of standard controls for popovers and the tendency for information to appear far away from the associated button became a huge barrier to efficient navigation of this app. Users relying on screen readers may not know whether new content loaded, whether the part of the page that changed is the desired section, or, in the case of unlabeled buttons, what the button does and where they should expect to go after they press it. In many cases, each new button required exploration of the entire screen, line by line, just to find out whether new content had loaded.

## Searching for Books

This applies for ebooks and audiobooks. Libby's *search results* and *loans* sections have dozens of blank lines separating the top of the page from the main content. These lines are skipped by some screen readers, but NVDA users are forced to arrow through them.

With both audio and ebooks, the screen to play or view them is completely blank except for the title of the book and the usual *library* and *shelf* navigation buttons. Screen readers cannot see any of the controls, time / page information, or book text. Books are effectively impossible to read or navigate with a screen reader using this app.

## Developer Recommendations

- Bring the Windows app's ebook and audiobook viewers in line with the ones found on other platforms.

- Eliminate the numerous blank lines from the interface, particularly in loans and search results.
- Consider troubleshooting the inability to navigate this interface using the tab key, as many keyboard users will prefer this.

## Conclusion

In this report we highlighted the main barriers of the Libby platform for readers with print disabilities. We also recommended some improvements to make it a better option for users relying on assistive technologies. The most important priorities across all platforms (iOS, Android and Windows) include: Label all buttons, controls, and links to convey information of what they are to users relying on a screen reader; use standard popover controls to eliminate backfield interference and confusingly mingled screens of information; allow more visual customizations, particularly the ability to independently change foreground and background colour; and make the text of ebooks accessible to screen readers.

We recognize and appreciate the changes made to the audiobook and ebook viewers on major platforms; these were the most problematic parts of the interface, and significant progress has been made since our last report. We hope that the recommendations in this document can help to facilitate further improvements throughout the Libby platform.

## Contributors

The following people contributed to this report, including testers and editors.

- Karoline Bourdeau
- Mélissa Castilloux
- Danny Faris
- Kaden Faris
- Simon Jaeger
- David Kopman
- Daniella Levy-Pinto
- Ka Li
- Deanna Ng