

Notes.net

Iris Today

Home

Download

Iris Today

Iris Cafe

All About Domino

Iris Sandbox

About This Site



by
[Susan Florio](#)

Level: Beginner
Works with: Notes 5.0
Updated: 05/03/99

Inside this article:

[The evolution of accessibility in Notes](#)

[Keyboard accessibility](#)

[Screen reader accessibility](#)

[Designing accessible applications](#)

Related links:

[Notes R5 Technical Overview](#)

[WebABLE](#)

[Microsoft Active Accessibility](#)

[Best Practices Guide](#)

[Bobby](#)

[Web Accessibility Initiative](#)

[Special Needs Web site](#)

Get the PDF:



access.PDF 42Kb



Notes is a tool that helps people find, create, and organize information. If you think about it, that is what many software products and Web sites do. But who are these products designed to help? Well, most of us would probably answer -- anyone who uses them. But when we say that, do we really mean it? What about people with disabilities? One of the main goals for R5 was to improve accessibility for the blind. Notes R5 is the first step towards making Notes a truly accessible product. If you are a blind user, you can now use the Notes client to do all of your normal day-to-day work. Let's take a look at how we achieved this goal.

Accessibility is defined as the quality of a system incorporating hardware or software that makes it usable by people with one or more physical difficulties, such as restricted mobility, blindness, or deafness. For the purposes of this article, we will refer to accessibility as it relates to blind users, although the changes we've made will help users with other disabilities. Blind users have historically had the most difficulty using the graphical user interface. In Notes R5, you can find accessibility improvements in two key areas:

- Keyboard accessibility
- Screen reader accessibility

In this article, we will show you what the improvements are in these areas and how they make Notes more accessible to blind and visually impaired users. We'll also show you how you can design more accessible applications. For an introduction to Notes R5 features, see the "[Notes R5 Technical Overview](#)". To learn more about disability-related Internet services, check out the Web site [WebABLE](#).

Understanding how blind users use Notes

Before you can understand how Notes is accessible to blind users, you have to understand how blind users use a computer. Blind users access software or Web pages using a screen reader. A screen reader is a combination of special software with a hardware or a software speech synthesizer. Winvision and JAWS are two examples of the software components of a screen reader. Both offer sophisticated software that translates complex user interfaces for blind users. This gives users the ability to read the entire screen, review just a line of text, or spell-check a highlighted block of text. A very popular example of a hardware speech synthesizer is Dectalk, which is also used to synthesize speech on many phone systems. If you have a hardware synthesizer, it sits on your desk and plugs in to your computer.

Cynthia Ice, Notes Accessibility Product Manager, recently showed me how her screen reader worked. As she used keys to move around the screen, the synthesizer read the text that appeared on the screen. I was surprised how quickly the words were read. Unless I really concentrated, I couldn't understand what the synthesized voice said. Cynthia explained that blind users get used to the speed and that it actually was on a very slow setting! She also pointed out that some blind users use a software synthesizer, which acts as a bridge between the screen reader software (such as JAWS or Winvision) and the sound card in the PC. The sound card itself verbalizes what is on the screen. She explained that when the screen

reader "reads" aloud the text from an application or a Web site, it reads from left-to-right and top-to-bottom, just as you would read a page. To be more accurate, the screen reader actually reads the underlying code of the application and translates that code to speech.

To get an idea of what it is like for a blind user to use a computer, imagine someone reading you a description of what is happening on the screen or under your cursor. For example, if you press CTRL+O, you might hear (in a suitably synthetic voice):

"Dialog box, Open Database. Server combobox, Database listbox, Filename text entry box. Pushbutton Open, pushbutton Bookmark, pushbutton Cancel, pushbutton Help, pushbutton About, pushbutton Browse."

As you tab through the dialog box, moving the keyboard cursor, you would receive verbal feedback on each control, such as:

"Server combobox, Local selected, Alice, Clapton, Dolly..."

"Database listbox, Cynthia's mail selected, Cynthia's mail, Local Address Book, Iris Address Book..."

"Text entry box, C, I, C, E, underscore, D, B, period, N, S, F"

If your cursor moves to a control that has no text label, for example, an icon or a custom control, you probably won't hear anything, or the screen reader sometimes just says "icon." A blind user might not even know that this icon even existed. If the icon is a key piece of navigation for the application, this can render the application useless to a blind user.

For many blind users, a screen reader is one of the most important tools they can have. It allows them to work at jobs where they use a computer and computer applications. It also opens up the vast world of information available on the Web. Unfortunately, it has become increasingly difficult for blind users to keep their computer-related jobs and their access to all the information available on the Web. This is because many new software applications and Web sites use a greater number of graphics, including toolbars and icons, than ever before. These UI enhancements lose their impact when they're only accessible to sighted users.

The evolution of accessibility in Notes

Notes R4.6 was the first time Notes added accessibility support. You could read e-mail, for example. However, it uses elements that a screen reader can't detect. For example, a screen reader doesn't know what a section is. So if you read a Domino Web site that uses sections with a screen reader, the reader just ignores the section, because it doesn't know it is there -- it doesn't recognize the code. Also, R4.6 is not completely keyboard accessible. This is a problem because even if a screen reader can read all the information in a product to the user, if the user can't type in keyboard combinations, it is impossible for the user to use the product. R4.6 did allow you to create accessible applications and Web sites with Notes, but you had to stay away from using any Notes elements.

R5 makes more Notes elements accessible to screen readers and it allows for greater keyboard accessibility. Certain elements, such as Java elements, are not accessible in R5. This is because a bridge between Java applets and screen readers is in its infancy. Also, group calendars, subscriptions, and the ability to create agents are some other areas of Notes that will not be accessible in R5. However, "From the beginning of R5 development, we have had the goal to release a 100-percent accessible Notes client," says Eric LoPresti, a UI designer and accessibility champion at Iris. "This means that anything that a sighted user can do in their mail, calendar, discussion database, and so on, a blind user can also do with the help of a screen reader."

Keyboard accessibility

The first thing we've done to make Notes more accessible is to extend keyboard access. Blind users generally don't use the mouse to navigate through Notes; they use the accelerator keys or command keys (such as CTRL+C or ALT+ENTER) on the keyboard. Accelerator keys allow you to use a mnemonic to trigger a button, control, or menu item. In Windows, accelerators are marked by underlining a letter in the label for that control, for example, F in the File menu. Command keys usually trigger an action, and often use the Control (or on the Macintosh, the Command) key. It is possible for blind users to use a mouse and most hardware synthesizers include the ability to use a mouse. However, when Cynthia showed me how a mouse worked, it was a very slow way for her to get information about what was on the screen and this made it almost unusable. The problem is that when keyboard access isn't available, a mouse is sometimes the only alternative. This is why in R5 we've provided accelerator keys for all menus, menu items, dialog box controls, and the search bar. The most highly used and important commands have command keys.

We've also made sure that keyboard navigation is consistent. For example, the following keyboard actions have these corresponding results:

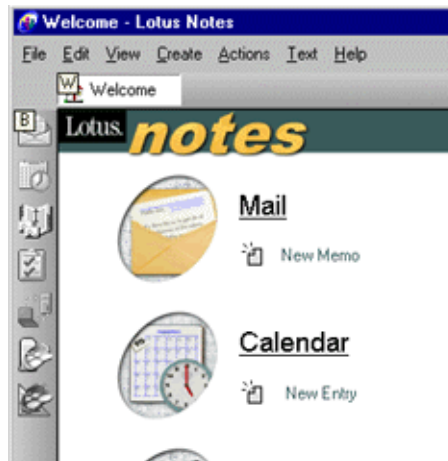
- Tab - navigates to the next control
- Shift+Tab - navigates to the previous control
- Ctrl+Tab - navigates between tasks
- Arrow keys - move the keyboard focus within a control or navigate to the next control
- Space bar - selects objects or items in a list
- F1 - opens context-sensitive help
- F6 - navigates across frames
- Enter - enters a value or activates a default action
- Escape - cancels or exits the dialog box

In addition, more controls have unique accelerators. There are two exceptions, which will not be keyboard accessible:

- SmartIcons
- The status bar

While you can't get keyboard focus for the status bar, a screen reader can read you the information that you could get from the status bar. For example, as you go through your mail deleting messages, the screen reader announces the number of messages you delete. It also periodically says, "Checking for new mail."

Keyboard accessibility isn't just for blind users -- sighted users can benefit from this kind of accessibility as well. One problem with accelerator keys is that they can be hard to remember unless they are visible on the screen. For example, it can be hard to remember that ALT+B brings you to the Notes Bookmarks bar, or that ALT+W brings you to the open page (known as tasks in R5) shown in the tabs along the top of your screen, because nowhere on the screen can you see a reference to this accelerator. The user interface designers at Iris provide an elegant solution to this difficult problem. In R5, when you press and hold down the ALT key, small yellow boxes with the accelerator keys inside of them appear near the items they refer to. This can be especially helpful when you use an accelerator sequence or more than one key. The following screen shows the yellow boxes that appear when you first press the ALT key.



Naturally, this feature is completely accessible to a screen reader, so in place of seeing yellow boxes, the blind user hears a list of options: B for bookmarks, W for Windows, and so on. This lets any user, blind or sighted, navigate to any bookmarks page quickly and easily, without having to remember the accelerator key they need.

Screen reader accessibility

The second thing that makes Notes more accessible is that it is now screen reader accessible. In order for a screen reader to work, it must receive detailed information about the GUI. It then translates the graphical display into speech. For this reason, screen readers generally work best when the "speaking" controls are native to the operating system. Notes 4.6 does not work well with screen reader software, in part, because it uses so many custom controls. In R5, we've made the client accessible using [Microsoft Active Accessibility](#) (MSAA).

MSAA runs on Windows 95, and provides a generic method by which screen reader software can get GUI information from Notes. However, this means that screen reader vendors must also make their software interface with MSAA. Here at Iris, we currently work with several screen reader vendors. We provide them with beta code and identify where we use MSAA and what MSAA code Notes uses to expose information.

Microsoft developed MSAA as an enabling technology to help make software more accessible to disabled users. If developers want to create accessible applications, they have to first make sure every action is accessible through the keyboard. Then they have two options. They can:

- Use standard Windows controls
- Include MSAA to expose information about non-standard controls

In Notes, we chose to incorporate MSAA in order to make it easier for screen reader vendors to enable their products to work with Notes.

As you move up and down a list of documents, a screen reader tracks the documents you highlight. As a result of using MSAA, the screen reader announces the entire document title line, as well the status of the document (such as, deleted, unread, or selected). In a discussion database, it can tell you the title of a document and whether there is a response document. In your mail file, it tells you that you deleted a message, whether or not you have read a message, and so on. MSAA handles all of this. We've also MSAA-enabled the Editor so that a screen reader can step through the document properly. In addition, views, the View and Folder pane, the workspace, and the status bar are all MSAA-enabled. Each screen reader vendor handles this differently and uses different keyboard commands to allow users to move through a document, but the important thing for blind

Notes users is that this ability is there. Any screen vendor that uses MSAA has a product that can work with Notes.

"Iris Associates is the first major software company, other than Microsoft, to implement MSAA," says Ice, "and according to several screen reader vendors, we have a better implementation of MSAA than Microsoft. This gives us a competitive edge." In R5, everything that you can do in Microsoft Outlook, you can also do in Notes. Implementing MSAA involved re-architecting Notes, and R5 provided a tremendous opportunity to implement something of this scale.

Designing accessible applications

You can make most objects you create in Designer accessible. For example, images can have alternate text tags (using Alt tags). Failure to provide alternate text tags is one of the most common reasons that many Web applications are not accessible. Also, many Web sites include text-only versions of their content.

It is imperative that you are aware of the accessibility consequences of your design decisions. For example, you might want to place a graphical icon in your application. Normally, a screen reader cannot read an icon, and thus, your application is inaccessible. However, if you follow our [Best Practices Guide](#), you will fill in the Alt text field, which labels the icon with a piece of descriptive text, such as "Click here to go to the next page." Then, when a screen reader comes to the icon, it reads the text aloud to the blind user and this icon becomes accessible.

To make sure your application is accessible, follow these guidelines:

- Avoid using Java applets. Java is not yet very accessible, and most browsers do not support keyboard navigation to applets on a page. If you need to use a Java applet, make sure to fill out the Alt text field in the Applet Properties box, and try to provide an alternative way to do the same action elsewhere.
- Avoid using color or graphics as the sole way to convey meaning.
- Fill out the Alt text field for all graphics. This field is found on the Basics tab of the Picture Properties box.
- Avoid excessive use of framesets and embedded objects, which can make keyboard navigation difficult.
- For every action button, make sure that a corresponding item appears in the Action menu (in R5, this happens by default). In R5, action buttons are only partially accessible. Including actions in the Action menu is a good practice, which also makes things easier for sighted users.

Notes R5 allows you to create an accessible application, but as a designer, you must take on the responsibility of keeping accessibility in mind when you design your applications.

Here at Notes.net, we plan to take advantage of what's coming in R5. Currently, our site is partially accessible. As we move the site to R5, we will design our applications to take advantage of new accessibility features. Even today, we strive to make our site accessible by providing alternate text for as many graphics as we can. In the future, we hope that the use of HTML 4.0 will enhance the accessibility of our site.

If you want to test your own Web site to see how accessible it is, you can use [Bobby](#), a free tool from a company called CAST. Bobby lets you know what parts of your site are inaccessible and rates your Web site overall on accessibility.

Conclusion

"Time is always a factor," says Ice, "but we are working really hard to make Notes accessible in R5, and we have already improved the product substantially in this area. Of course, we will continue to add improvements geared towards blind users in future releases."

As we move into the future, we expect that the W3 will impact much of what we do. The W3 is committed to making the Web available to people with disabilities and they have been setting standards and passing laws regarding this. They have launched a [Web Accessibility Initiative](#) (WAI) with recommended standards that make the Web more accessible. Among these recommendations is the use of HTML 4.0 tags, which provide more accessibility to blind users. Most Web professionals believe that in the future, the use of style sheets by Web developers will make the Web more accessible. However, style sheets and HTML 4.0 are still new to the market and the large majority of Web application don't yet use them.

IBM is also committed to accessibility for all of their products. They're working closely with the W3 in this effort, and their [Special Needs Web site](#) provides additional information on accessibility.

Our goal of making Notes more and more accessible is right in line with the goals of these organizations, and R5 is proof that we are following through on our commitment.

AUTHOR'S NOTE

Susan would like to thank Antonio Robinson, Notes Client Product Manager, and Cynthia Ice, Notes Accessibility Product Manager, for all their help with this article.

What do you
think about
this article?

Register
Here!

[Lotus Home](#) | [IBM Home](#) | [Iris Home](#) | [Feedback](#)
Copyright 1999 Iris Associates Inc.

