

UTILIZING LOTUS SYMPHONY TO CREATE OPEN AND ACCESSIBLE DOCUMENTS

Authors:

Brian J Cragun, IBM, cragun@us.ibm.com

Susann Keohane, IBM, skeohane@us.ibm.com

Xing X Li, IBM, lixxing@cn.ibm.com

Description: Techniques and strategies for creating and testing accessible documents in open format using Lotus Symphony.

Table of Contents

UTILIZING LOTUS SYMPHONY TO CREATE OPEN AND ACCESSIBLE DOCUMENTS.....	1
Introduction.....	2
The OpenDocument standard.....	2
Background and Accessibility Design of Symphony.....	3
Symphony's Extensible Design.....	4
Accessibility Enhancements.....	4
The Navigator.....	4
Dockable Sidebar.....	5
User settings.....	6
Creating Accessible Symphony Documents, Presentations and Spreadsheets.....	6
Language setting.....	6
Text formats and color usage.....	7
Data tables, row and column headers:.....	8
Images, graphs, and charts.....	10
Reading order.....	11
Navigation.....	12
Text document navigation.....	12
Presentation navigation.....	13
Spreadsheet navigation.....	13
Range names.....	14
Creating other accessible outputs.....	15
Testing Symphony content for accessibility.....	15
Test guidance.....	15
Phase 1—Preliminary quick check to ensure development techniques were followed.....	15
Phase 2—Use automated tools, if available, to quickly scan for accessibility.....	16
Phase 3—Perform all manual testing.....	16
Phase 4—Test with a screen reader.....	16
Phase 5—Accessibility problem analysis.....	16
Automated Tools – ACTF aDesigner.....	17

Visual inspection.....	18
Manual test.....	19
Verify reading order and document navigation for Symphony Documents.....	20
Verify reading order and document navigation for Symphony Presentations.....	20
Verify reading order and document navigation for Symphony Spreadsheets.....	20
Test with a screen reader.....	21
Summary / Conclusion.....	22
References.....	22

Introduction

IBM® Lotus® Symphony™ is a next generation office application that is strategically committed to openness and choice by enabling creation of accessible documents in open formats as well as a variety of proprietary document formats. It can be used to create accessible text documents, presentations and spreadsheets. Using IAccessible2 (IA2) technology, an extended accessibility application programming interface (API), it provides a full set of accessible navigation capabilities, including assistive technology (AT) access to advanced functions such as styled text, tables, spreadsheets, lists and drawing objects.

This paper first presents the background and accessibility design of Symphony. It provides strategies and techniques for creating accessible content (text documents, presentations, and spreadsheets) with Symphony. It then explains strategies for verifying that content created with Symphony is accessible.

The paper assumes no prior knowledge of Symphony, OpenDocument format or accessibility of Symphony content. The testing section assumes a basic exposure to using a screen reader. This paper is based on Lotus Symphony version 1.2.

The OpenDocument standard

OpenDocument Format is an open standard XML-file format specified by OASIS (Organization for the Advancement of Structured Information Standards). It is also published as a joint ISO (International Organization for Standardization) and IEC (International Electrotechnical Commission) International Standard. Documents created in OpenDocument Format are compatible with many applications and can be imported and exported to many formats. An advantage of the OpenDocument format is its open nature, allowing it to be documented, viewed, and understood in its raw state, and keeping it free from private control. Therefore, users creating content in OpenDocument format are able to choose from multiple vendors when creating and viewing content.

OpenDocument format supports text document, presentation, and spreadsheet content. The common file types associated are .ODT for text documents, .ODP for presentations, and .ODS for spreadsheets. OpenDocument Format is compatible with current and future versions of Symphony, as well as the OpenOffice application suite.

Because ODF is an open standard, projects are under development throughout the accessibility community to create automated checking and analysis tools. As time goes on, these building and checking tools will increase in number and sophistication. We anticipate that ODF accessibility tools will eventually exceed the number of accessibility tools available for proprietary document formats.

Background and Accessibility Design of Symphony

Symphony is based on OpenOffice.org and uses Eclipse™ as its front end to provide an integrated set of productivity applications. Symphony has made many usability, interoperability and accessibility enhancements on the OpenOffice.org code base to satisfy the productivity tool requirements of persons with disabilities (PwD).

Ease of use was a primary objective and was necessary to achieve the accessibility compliance requirements. Symphony provides full keyboard access, visual focus and high contrast for all functions, including dialog controls and editing areas. The user interfaces of some key features were redesigned to increase usability.

In addition to usability enhancements, there are many improvements in document interoperability with other document formats, such as Microsoft® Office, Lotus SmartSuite® format and Portable Document Format (PDF). All information required for accessibility is maintained when opening and saving documents, thus avoiding any accessibility loss during document conversion.

Accessible information is critical for people with disabilities (PwDs). Assistive technologies such as screen readers must rely on standard APIs to acquire the accessible information from applications. The Microsoft Active Accessibility® (MSAA) API is the most popular API on the Windows platform. It was released in 1995 and supports dialog controls and menus. Yet, MSAA does not support rich text, tables, lists and other widgets that are commonly used in modern productivity software and Web applications. Because of these limitations, ATs must resort to API hacks and application-specific behaviors to get the desired information. Even worse, some applications have added specific private interfaces for AT access. Such a model is a huge burden for both applications and ATs to maintain.

By merging and extending the virtues of different accessibility APIs, a new accessibility API IAccessible2 (IA2) was created. IA2 is an open standard, developed by IBM and donated to the Linux® Foundation. It builds on the existing MSAA architecture by utilizing its Active Accessibility COM interface and the event generating mechanism and extends them to support many other document elements, such as table, text, hyperlink, and more. This allows applications to continue using their current implementation of MSAA and make only those enhancements needed to support IA2.

Lotus Symphony is the first office application suite to implement IA2. Developers worked closely with AT companies to ensure a complete solution and achieve the accessibility requirements of PwDs for both document creation and review. MSAA and IA2 together parallel the scope of the Linux-based Assistive Technology Service Provider Interface (AT-SPI). By implementing IA2 along with MSAA, Symphony is able to identify relationships between labels for controls, headers for table rows and columns,

tree nodes, tree node expansion status, as well as nesting levels for trees and lists.

Symphony's Extensible Design

Lotus Symphony is built on an Eclipse base and has a flexible plug-in design. Many Symphony-related third party and community built plug-ins are becoming available. Accessibility enabling tools which build and check ODF content can be integrated directly into the Lotus Framework. For example, IBM has demonstrated an early prototype of a Symphony-integrated tool to check and repair ODF files.

Symphony's framework also allows it to be embedded within other Eclipse frameworks. For example, Lotus Notes 8.5 can host Symphony so that ODF documents can be viewed in an integrated fashion in Notes tabs.

Symphony supports the ability to import and export several file formats. Symphony imports and exports Microsoft Office formats up to Office 2003. It also exports PDFs, retaining accessible content created through Lotus Symphony.

Symphony operates cross-platform on Microsoft Windows[®], Linux, and Apple OS X. Symphony's accessibility features are tested on Windows.

Accessibility Enhancements

Symphony provides several usability enhancements to achieve accessibility as an application. These build upon the foundation of IAccessible2, already discussed. These are the Navigator, dockable sidebar properties, and user controls

The Navigator

Symphony provides an easy way to organize and select graphical objects on the screen without a mouse. The *Navigator* view is shown by selecting menu *View>Navigator* or pressing *Ctrl + Shift + F5*. It shows an organizational summary of all the objects in the text document, presentation, or spreadsheet.

The user has a quick overview and access to all the objects by type. For example, the user can see a list of all the images, tables, or hyperlinks in the file. The navigator shows the text of headings in a text document as the text of hyperlinks. Most objects in the Navigator allow access to their properties dialog through a context menu. Selecting an object in the Navigator will adjust the editor display so the object can be seen.

The *Navigator* is a powerful tool for those unable to use a mouse because it allows selection of one or multiple objects for keyboard menu interaction. To select elements without a mouse, use the keyboard to move the highlighter bar to the desired node or object, and then press the Spacebar to select it. Repeat the process to select multiple items. If the node is already selected, highlighting and pressing the Spacebar deselects it. Selected nodes are indicated in the Navigator with an "*" next to the name.

Figure 1 shows an example of using the Navigator in a presentation editor session. Each object on the current slide is shown in the tree structure of the *Navigator*. The currently selected object on the slide is highlighted with green handles and simultaneously is shown in the *Navigator* with an "*" next to the name.

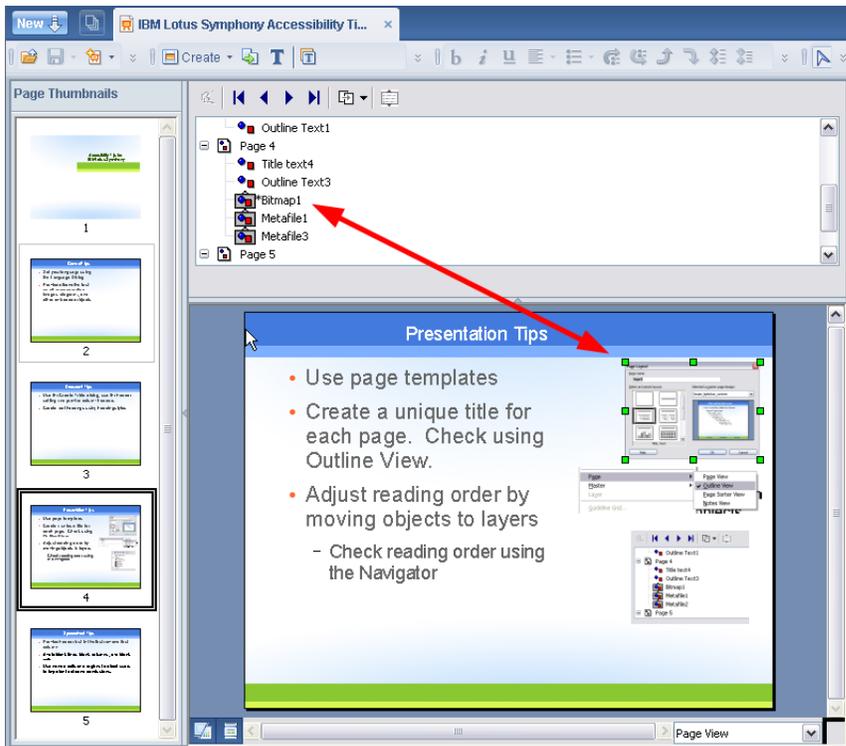


Figure 1– Symphony presentation editor with the thumbnail, preview and Navigator panes displayed. A bitmap is selected in the Navigator and the corresponding image is highlighted in the slide preview.

The objects shown in the Navigator vary for each type of Symphony file. They are shown in Table 1.

<i>Content type</i>	<i>Objects shown in Navigator</i>
Text document	headings, tables, frames, graphics, notes, objects, sections, indexes, and hyperlinks
Presentations	pages, title frames, text frames, images, objects, and tables
Spreadsheets	cells, sheets, range names, linked areas, graphics, notes

Table 1 - Objects shown in the Navigator by content type

Dockable Sidebar

In addition to the standard Properties dialog reached through the context menu, properties may be shown dynamically for each currently selected element in a sidebar next to the editing area. The properties in the sidebar represent the most commonly used and accessed property elements. When typing text for example, the font, effects (e.g.

Strikethrough) and position (including rotation) of text at the current cursor position is shown. As other objects are selected, the basic properties are shown. The number of keystrokes required to change these properties remain approximately the same, but for those with limited mobility, the ability to see these settings without continuously using the context menu saves significant time and interaction effort. Figure 2 shows a text document editing session with the sidebar listing the current properties of the selected text. The properties are active and updates are shown dynamically in both the properties and the displayed text document.

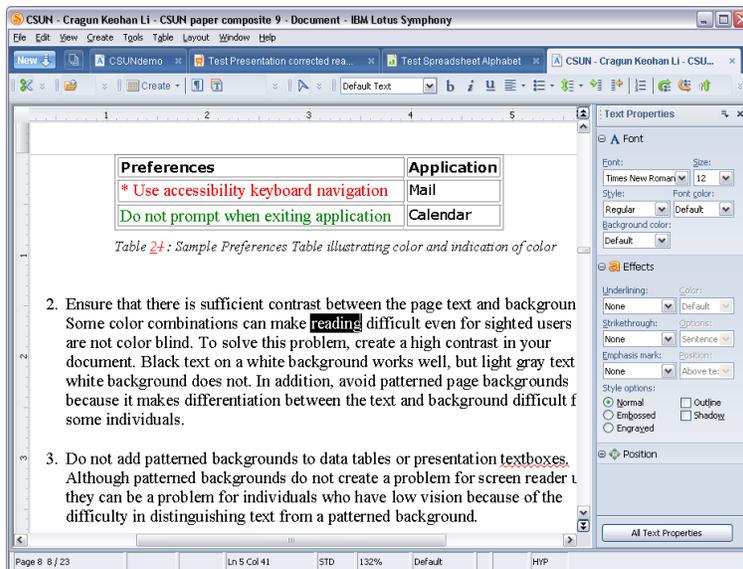


Figure 2 – Sample text document editing session showing text properties in the sidebar.

User settings

Settings are provided through the preferences dialog for several common user accessibility needs. A setting is provided to allow a text selection cursor in read-only text documents. Another setting turns off all animate graphics and text. The user can also control the time that help tips remain visible on the screen.

Creating Accessible Symphony Documents, Presentations and Spreadsheets

The next sections provide techniques to help authors create accessible content using Symphony. Except where noted, the techniques apply to all Symphony content: text documents, presentations and spreadsheets.

Language setting

It is important to identify the primary language of the content (document, presentation or spreadsheet). The language setting is used by the AT to correctly read the content. In Symphony, use the *Preferences* dialog (*File > Preferences*) to define the primary language of the document.

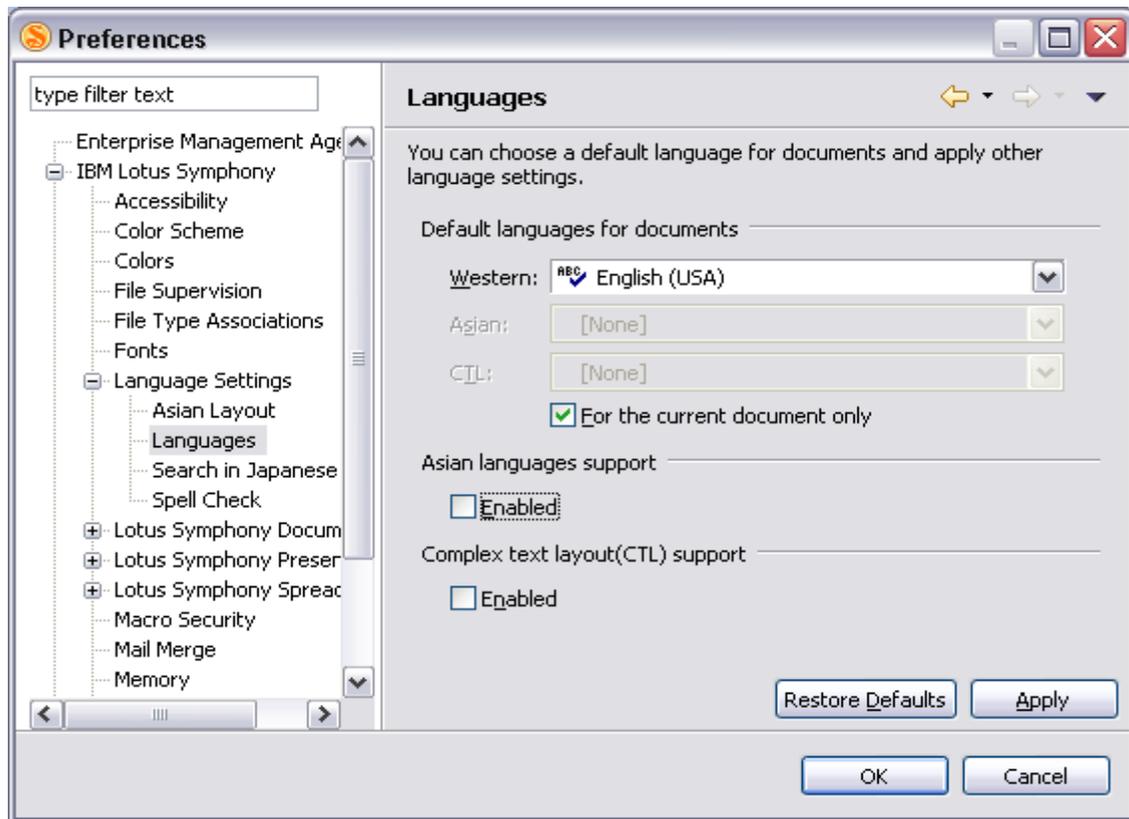


Figure 3 – Preferences dialog

Text formats and color usage

Screen readers may have difficulty reading text formatting or detecting text color. As a result, the screen reader will read the formatted text as if it contained no formatting. For users who are color-blind or visually impaired, the use of the colored, underlined, and bold formatting fails to convey the intended information. For all users to understand and gain the full benefit of a document, text formatting and color may be used as an enhancement, but not as the only way to convey information.

To create documents, presentations and spreadsheets with accessible text formatting and use of color, follow these techniques:

1. Use text formatting and colors for enhancement, but not as the only means to convey information. If you choose to use text formatting or colors to convey information for the sighted, provide the information in a redundant form so that color blind, low vision, or screen reader users can differentiate it.

For example, if required user settings are listed in red and optional settings are displayed in green, preface the setting with an asterisk or other cue so someone who is colorblind or someone who is using a screen reader will also be able to identify the required commands.

*=Required settings

Preferences	Application
* Use accessibility keyboard navigation	Mail
Do not prompt when exiting application	Calendar

Table 2 : Sample Preferences Table illustrating color and indication of color

2. Ensure that there is sufficient contrast between the page text and background. Some color combinations can make reading difficult even for sighted users who are not color blind. To solve this problem, create a high contrast in your document. Black text on a white background works well, but light gray text on a white background does not. In addition, avoid patterned page backgrounds because it makes differentiation between the text and background difficult for some individuals.
3. Do not add patterned backgrounds to data tables or presentation textboxes. Although patterned backgrounds do not create a problem for screen reader users, they can be a problem for individuals who have low vision because of the difficulty in distinguishing text from a patterned background.

If your document contains colored text or objects, there is an easy way to verify that the information being conveyed is not color-dependent: print or view the document in black and white -- the information in black and white should be no less meaningful than a color version.

Data tables, row and column headers:

Users must be able to understand the purpose of a table, and understand the purpose of the data contained within the table. When a user of assistive technology navigates through the cells of a data table, the user must be able to determine the column or row header in order to understand the meaning of the data contained in it. Ensuring data tables are accessible requires techniques unique to each Symphony format (e.g., text documents, presentations, and spreadsheets).

Do not use tabs or spacing to create tables. While it may visually look like a table, it will not be recognized as a table by assistive technology and will not be accessible.

For text documents, Symphony enables the user to set both the row and column headers. To create accessible data tables:

1. Using the *Table > Create Table* dialog for a new table, select the *Column Header* and *Row Label* checkboxes to specify column and row headers of a data table as shown in Figure 4a.
2. For an existing table, set the row and column headers on the *Rows and Columns* tab of *Table Properties* dialog, via the *Table > Table Properties > Row and Columns*

tab as shown in Figure 4b. Check the *Column header* and *Row label* checkboxes and provide the number of rows and columns that make the table headers. Because screen readers cannot read table headers that span more than one row or column, verify the number of rows and columns are not greater than one.

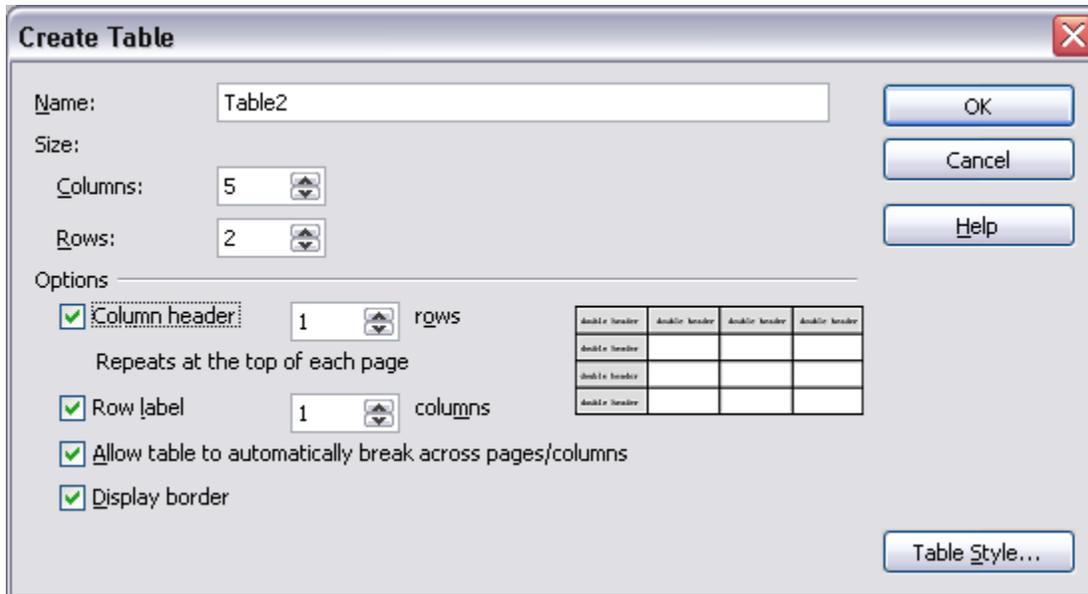


Figure 4a – *Create Table* dialog showing the column header and row label checkbox fields

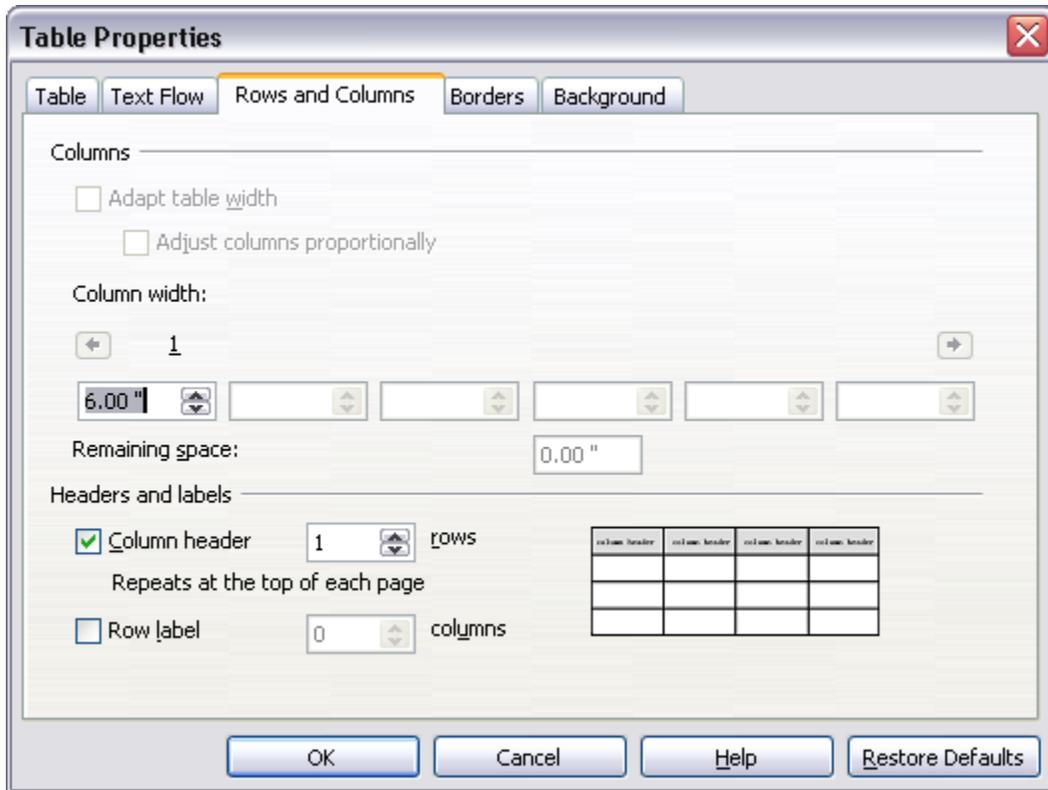


Figure 4b – *Table Properties* dialog showing the *Column header* and *Row label* settings on the *Rows and Columns* tab

3. Provide informative header text that conveys the meaning of the data contained in the rows and columns. Brevity is appreciated because the text will be repeated many times.
4. Add a caption above or below the table to summarize the purpose of the table. Select the table by placing the cursor in a cell of the table then select *Create > Caption* to add the caption. Alternatively, add a caption by right-clicking anywhere on the table and selecting *Add Caption*.

For presentations, setting column headers and row labels is not yet an available option. To create accessible data tables:

1. Create simple tables using *Create > Table* which are accessible without any modification.
2. Add a caption above or below the table to summarize the purpose of the table.
3. Do not use patterned backgrounds within tables.

For spreadsheets, screen readers may not be able to interpret large data sets or ranges that are not labeled properly. If row and column headers are not labeled properly, users might not know what type of data is expected. To help ensure that data tables in the spreadsheet are accessible, use row and column headers extensively and avoid ambiguity within these headers. Make them clear, concise and self-explanatory.

To set the *Row* or *Column* title:

1. Select the titles you want to define,
2. Select *Create > Name > Label*. The *Create Labels for Names* dialog box will appear.
3. In the first editable field confirm the range of cells to be used as the header label.
4. Make sure the proper radio button is selected, either *Contains row labels* or *Contains column labels*.
5. In the *For data range* editable field confirm the range of cells that the header will label.
6. Select *Add*.

Images, graphs, and charts

In addition to text and tables, images and other visual objects may be included in your Symphony text documents, presentations and spreadsheets. To help ensure the context of images is conveyed to blind or low vision users, meaningful text equivalents or “alternative text” must be provided. Alternative text (alt text) is not needed for images used for visual effects only, but is required for all objects that add meaning to your document.

To add alt text for an image:

1. Right-click the image.
2. Select *Graphic Properties* from the menu.
3. When the *Graphic Properties* dialog box is displayed, select the *Alternative Text* tab.
4. In the *Alternative Text* field, enter a text description of the image. The alternative text should effectively describe the image so a sighted user and a blind user would get the same information from the image.

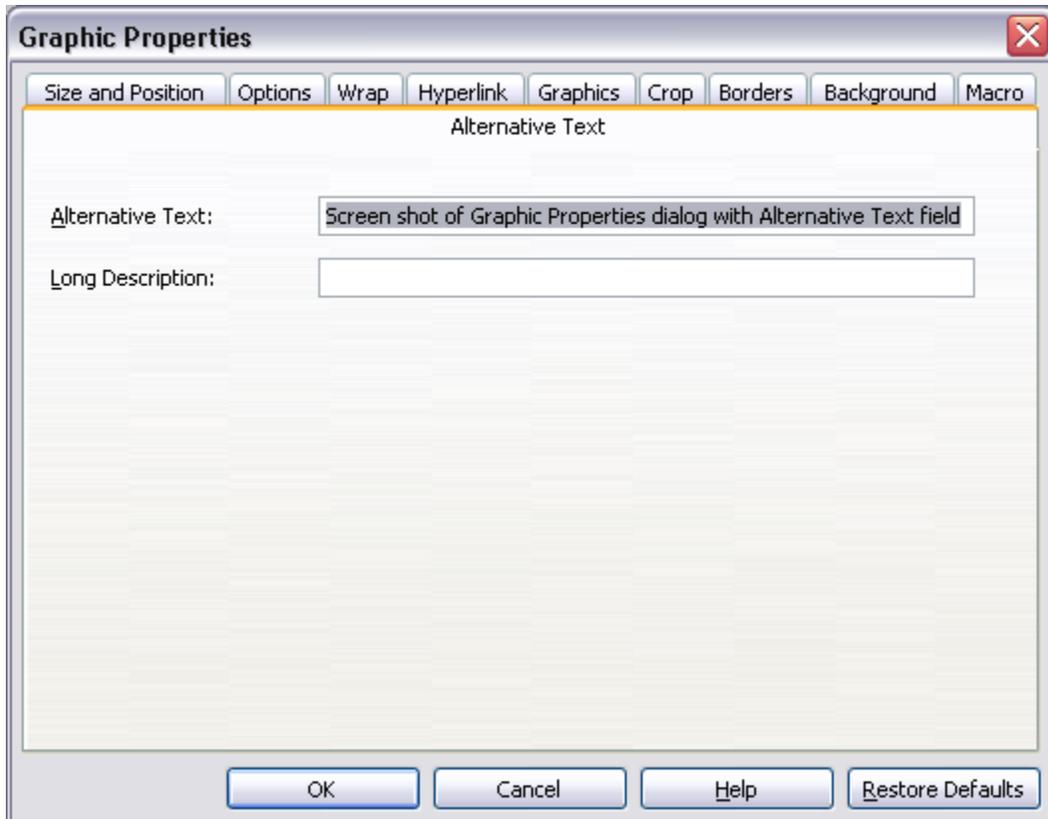


Figure 5 – *Graphic Properties* dialog showing the *Alternative Text* field on the *Alternative Text* tab

Within presentations, it is often necessary to group smaller objects into a single object. When an image is a composite of many elements, it makes the page difficult to read with a screen reader. Use the *Symphony Group* option to group many objects into a single object. To do this, select all objects in the group, then select *Layout > Group > Group*. Be sure to add appropriate alt text to the grouped image.

Reading order

Correct reading order is fundamental to creating accessible content. Content objects must always be placed in a logical reading order. This helps the document, presentation or spreadsheet flow correctly when using assistive devices. Simple layouts are recommended when possible. Complex layouts (such as found in brochures), sidebars, images wrapped by text, and other ornamentation may take additional work and testing to ensure a logical reading order.

For presentations, screen readers by default will read objects on a slide objects from back layer to front layer. Each time an object is added to the slide it is placed in a more forward layer. Consequently most slides do not end up in their most logical reading order. The order shown in the *Navigator* represents the back-to-front layer order in the file, and indicates the order in which layer objects will be read.

To explicitly set the reading order for a page use the following steps:

1. Select the first item to be read. (This can be done with the *Navigator*.)
2. Use the context menu to select *Arrange > Bring to Front*.
3. Move to the next to item to be read and follow the same process until all objects on the page have been ordered.

The last item selected with *Bring to Front* option will be the last item spoken on the page. Symphony reads objects in a slide from back-most layer to front-most layer. Remember to also set the order for page titles, sub-titles, and tables.

Symphony also provides the ability to set the keyboard navigation order for the objects in the presentation using menu item *Presentation > Object Navigation Order*. This dialog controls the tab key navigation for the objects, but does not change the reading order, because the reading order is based on the presentation layer of the object. Use of the *Arrange* context menu item is recommended, instead.

For spreadsheets, do not use blank cells for layout or formatting purposes. It's better to densely pack the data in the workbook and then use Symphony layout techniques. Avoid the use of white space with lots of blank cells or blank rows and columns. Screen reader users must make extra effort to skip over these cells.

Navigation

Users that do not use a mouse must navigate documents in a logical order. Screen reader users must read documents in sequential order unless authors provide an accessible method to navigate the documents. Ensuring screen reader users will be able to easily navigate the text document, presentation, and spreadsheet requires techniques unique to each Symphony format.

Text document navigation

For text documents, organize the information in sections with descriptive headings. Use the heading styles available in Symphony ("Heading 1," "Heading 2," etc.) to indicate the document structure. This enables screen readers to create an outline of the document and make the outline available to the user for navigation. Using bold text to indicate headings is not sufficient to create an accessible document. For long documents, provide a table of contents in addition to using heading styles. This improves a screen reader user's ability to navigate to specific sections of the document. Symphony provides the ability to create a Table of Contents using document Heading styles with little additional work.

To create a Table of Contents using predefined headings:

1. Select the placement of the Table of Contents within the document by placing the cursor at that location.
2. Select *Create > Index and Table > Indexes and Tables* to launch the *Insert Index/Table* dialog and then select the *Index/Table* tab.
3. Select *Table of Contents* in the *Type* drop-down list box.
4. Specify the document levels and style to be included in the table of contents by selecting the *Additional Styles* check box in the *Create from* group, and then select the "..." button next to the check box. In the *Assign Styles* window, select

- the style in the list, and then select the “>>” or “<<” buttons to define the outline level for the paragraph style. Once the list of styles to be included in the table of contents is complete, select OK.
5. Select OK again to create the table of contents.
 6. If content is added later to the document, use either of the following methods to update the table of contents:
 - a. Select the Table of Contents then right-click and select *Update Index/Table*.
 - b. Select *Tools > Update > All Indexes and Tables*.

Presentation navigation

For presentations, use the Symphony page layout templates and placeholders to add content to the page. Text added to the page using predefined templates and placeholders is automatically added to the outline view of the presentation. The outline view (*View > Page > Outline view*) enables users to quickly navigate a long Symphony presentation. Text that is inserted using *Create > Text Box* will not be part of the outline view and may not be read by a screen reader.

Under user control during a presentation, navigation between elements follows the same flow as the reading order, as discussed previously. Users reading a presentation navigate a table under direct control using *Ctrl + Alt + (arrow keys)*.

Spreadsheet navigation

For spreadsheets, providing navigation and orientation information maximizes accessibility and usability. Keyboard only (non-mouse) users and AT users need the ability to navigate directly to specific sections of a spreadsheet, without having to traverse an entire document. To enhance productivity for users of assistive technologies, enable navigation to specific sections of a spreadsheet by using the following techniques.

1. Organize the spreadsheet by providing basic navigation information and key highlights.
2. Define ranges with meaningful names to indicate important sections of the spreadsheet. More details on range names and how to create them follow.
3. Use descriptive text to explain what is in the spreadsheet. Embed this into the sheet by creating a range name called “information” or “instructions” that a user can easily navigate to and read.
4. Place the basic instructions on how to navigate the form into cell A1. Sighted users will see the instructions as soon as the sheet opens, and screen readers will read the instructions to users. It is also helpful to store instructional information and actual data information and computations in separate sheets.
5. On every sheet, provide a standard message such as “End of sheet” to let the user know there is no more data beyond this point.

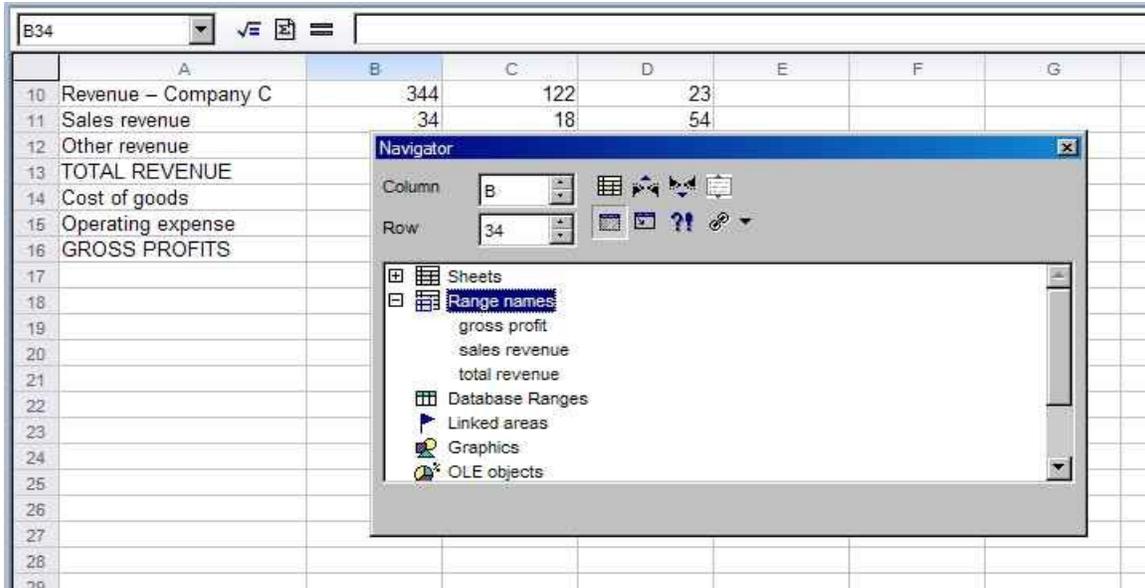


Figure 6 – Symphony worksheet with defined Range names shown in the *Navigator*

Range names

Range names provide navigation by informing the user that there are multiple ranges in the worksheet and the range names make it easier for a user to navigate to them. A list of range names is provided using Symphony's Navigator command (*Ctrl+Shift+F5*).

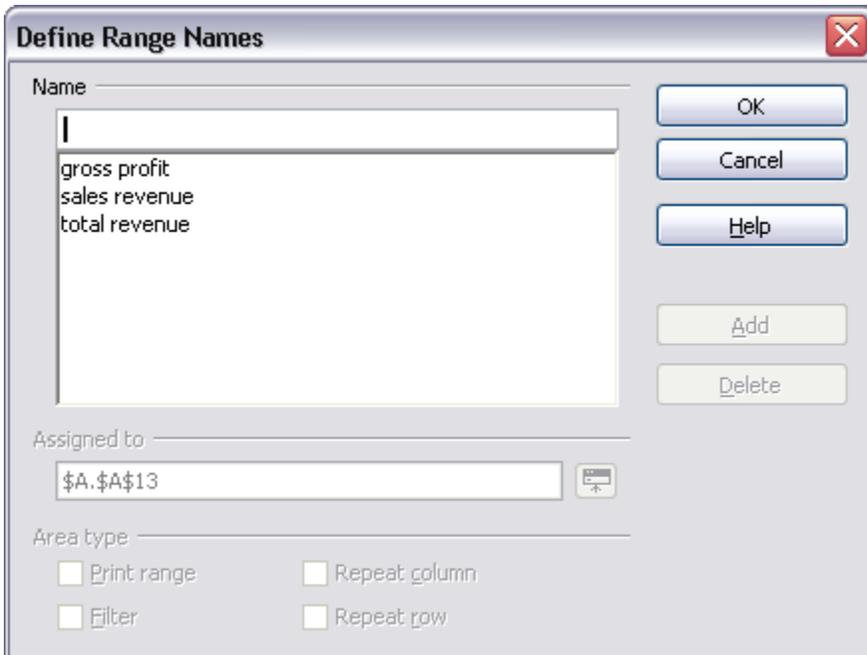


Figure 7 – *Define Range Names* dialog

To define a range name:

1. Select the range on the spreadsheet.

2. Select *Create > Names > Define* (or *Ctrl+F3*). The *Define Range Names* dialog box appears (as shown in Figure 7.)
3. In the first editable field, provide a meaningful name for the selected range.
4. Select *OK* or press *Enter*.

Creating other accessible outputs

Text documents, presentations, and spreadsheets can be saved as Portable Document Format (PDF) files. Content saved in this format retains the accessibility features added to the ODF file, such as language setting, alternative text for images, and reading order. The exported document is marked as a tagged PDF document. Tagged PDF files enable assistive technologies to recognize defined document structures such as paragraphs, lists, headings, tables and other structural elements.

Text documents can be saved in Microsoft Word® document format. Documents saved in this format retain sufficient information for most constructs to be accessible when read in Microsoft Word. Because Word documents only support row headers, all column header information is lost when converted to Word format.

Presentations can be saved in Microsoft PowerPoint® format and retain sufficient information for most constructs to be accessible when read in Microsoft Power-Point. Table information is not retained. Text documents and presentations saved in Microsoft Office formats must be tested using standard testing procedures before certifying them as accessible.

Testing Symphony content for accessibility

The following sections discuss testing files created with Lotus Symphony for accessibility. It first provides a recommended series of accessibility test phases and preparations necessary for accessibility testing. It then covers the various steps to test Lotus Symphony content. In most cases, the procedures are the same whether testing documents, presentations, or spreadsheets. However, when there are differences due to a specific document format, the differences in testing are noted.

Test guidance

Testing Lotus Symphony content for accessibility requires validating the content using visual, manual, screen reader, and automated tool techniques. Prior to presenting specific testing techniques, it is useful to understand five accessibility test phases. These test phases are listed in the most efficient order for testing, and make it easier to find accessibility problems at their earliest evidence and least costly point of detection.

Phase 1—Preliminary quick check to ensure development techniques were followed.

The development techniques already outlined in this document should be followed. Content authors should verify all development techniques are incorporated in the content using manual inspection and automated tools before proceeding to other phases of

accessibility testing. By doing so, teams ensure that accessibility problems are identified as early as possible. Testers are encouraged to work with the author to determine if the development techniques were implemented, and should spot check areas they suspect of not meeting accessibility compliance before progressing to the other, more formal testing phases.

Phase 2—Use automated tools, if available, to quickly scan for accessibility.

Automated tools give authors and testers the advantage of fast testing with broad coverage. Automated testing is limited to the rules that indicate the content is programmatically wrong; it is not able to verify contextual correctness. For example, an image can be tested using an inspection tool to determine if it has alternate text. However, the author and tester must view the image in context to determine if the alternate text provides a description of the image sufficient for assistive technology users.

Currently, there are few tools available to automatically check OpenDocument format files. One readily available example is the [Accessibility Tools Framework \(ACTF\) aDesigner](#). More details on how this tool can be used are outlined in the [Automated tools - aDesigner section](#) of this paper.

Phase 3—Perform all manual testing.

Manual testing requires the least setup and is the easiest to perform. All of the development techniques can be verified manually. It might seem faster to just jump into the screen reader test, but if something doesn't read properly, additional diagnostic work is required to find out whether it is the content, screen reader or a combination of both that is failing. Visual inspection is the easiest test. Keyboard navigation order is another easy test to perform. Other manual tests are made using the Lotus Symphony editor as described in detailed sections that follow.

Phase 4—Test with a screen reader.

A screen reader is software that reads the content and user actions aloud as the user navigates and interacts with the content. Most content can be tested for accessibility using manual techniques, however testing with the target screen reader application the only way to be certain what the user will hear. Screen reader testing is recommended for complex documents or documents with high use. When these phases are followed in order, virtually all defects will have been found before testing with a screen reader. It is especially valuable to test complex layouts to assure the content is presented in a way that makes sense to a person using a screen reader.

Phase 5—Accessibility problem analysis.

Accessibility problem analysis is sometimes required to determine compliance. It may be necessary to prove through the use of analysis tools that the content was developed correctly for accessibility and that a particular problem is caused by the editor or the assistive technology. This is an advanced topic that will not be discussed further in this paper.

Testing Lotus Symphony Content (document, presentation or spreadsheet)

In the following sections, this paper covers how to use these testing phases to test Lotus Symphony content. Testing Symphony is similar to testing other kinds of documentation, but there are a few differences. For example, the functional path to features in Symphony is different, and often there are different keystrokes to manipulate and edit the documents. Because Lotus Symphony is a relatively new product, improvements and fixes are frequently added to the screen reader rule-sets. Be sure to download the [latest version of Lotus Symphony](#). If performing a screen reader test, be sure to use the latest version of the screen reader.

It is important to verify the development techniques have been followed. The author of the content will know if he or she implemented the development techniques. Testers should verify with the author before proceeding to test. Both the author and the tester should review the development techniques and spot check any problematic areas they suspect of not meeting accessibility compliance before moving forward with detailed testing. It is highly recommended to begin testing only on content where development techniques have been followed.

Automated Tools – ACTF aDesigner

An initial step in testing Lotus Symphony content is to run an automated tool, such as the [Accessibility Tools Framework \(ACTF\) aDesigner](#). ACTF aDesigner is a standalone desktop application based on Eclipse that finds accessibility problems with images, tables, and headers in ODF documents. Upon selecting an ODF document, it performs several tests, and produces a report of warnings and errors. It also provides a score that can be used to measure compare documents and improvements made to a document. For each warning or error, help documentation suggests common fixes.

Fix each warning and error found by the automated checker. After automated checking is completed, continue on to manual checking of the content to assure its accessibility. All of the accessibility development techniques can be verified through visual inspection and manual test.

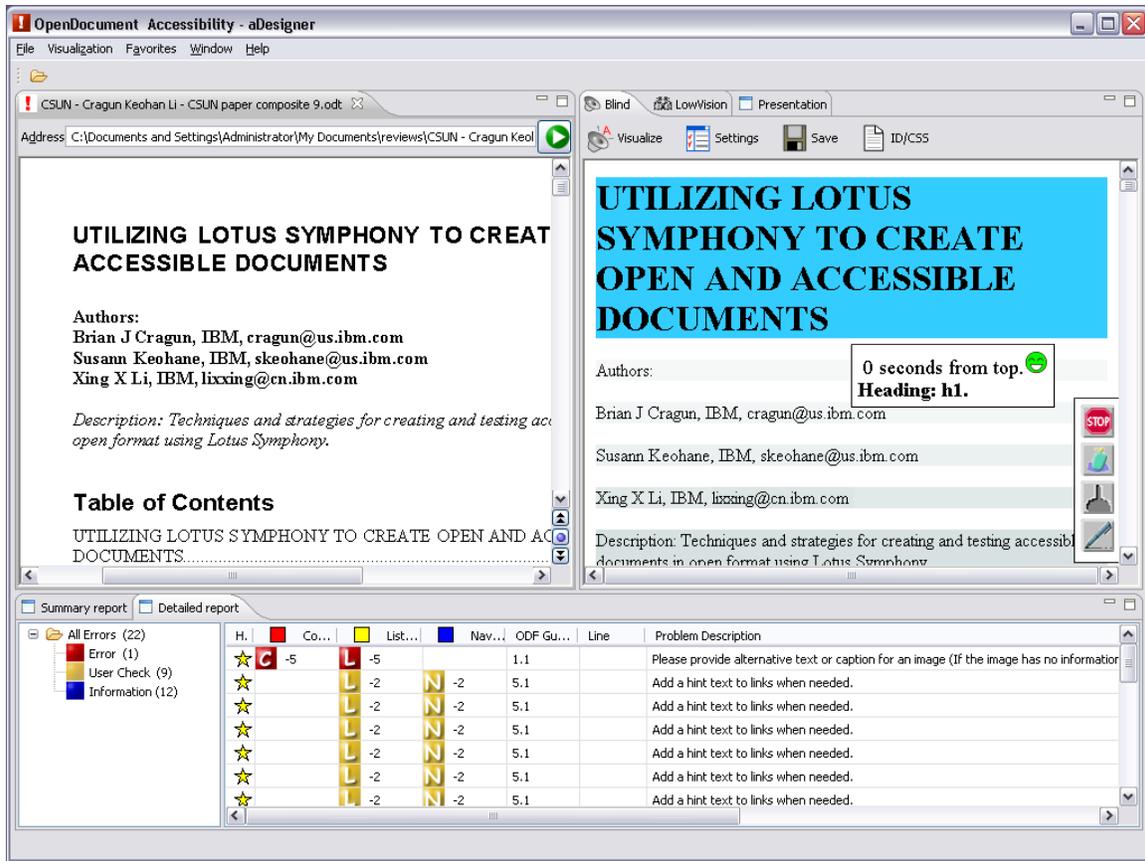


Figure 8 – aDesigner in ODF checking mode. Panes show (clockwise): the original document, an analysis view of the document as it would be heard by a screen reader, and a listing of all errors and warnings.

Visual inspection

To begin, review the document, beginning to end, looking for visible problems:

1. Watch for blinking or flashing text or images.
2. If color or text formatting is used, check that any meaning communicated through the use of color or text formatting is also communicated in another way. One way to check this is by printing a black and white copy of the document.
3. Check that there is sufficient contrast between the foreground text and the page background.
4. Check that when high-contrast and large fonts are used, that the text and other content in the foreground are clearly discernible from the page background.
5. Check links to ensure the text of each link is meaningful on its own (outside of context) and distinguishable from other links.
6. At this phase, it is helpful to make note of complex arrangements of text or

images in preparation for detailed checking outlined below.

Manual test

After an initial visual inspection of the content, verify all elements in the content are accessible using the manual test techniques described below:

1. Verify the language setting is defined.
 - a. Select the menu item *File-Preferences > IBM Lotus Symphony > Language Settings > Languages*.
 - b. Verify that the specified language in the *Default languages for documents* section matches the language of the document.
2. Verify images for alternative text and complexity.
 - a. For each image, verify the alternative text has been added by opening the *Graphic Properties* dialog box (*Layout > Properties > Graphic Properties*).
 - b. On the *Alternative Text* tab, read the text in the *Alternative Text* field to verify that the text sufficiently describes the image. Note: automated checkers usually find missing alternative text, but you must still verify text content.
 - c. In a presentation, images placed near each other with the intention to appear as one or an arrangement that is comprised of smaller images should be grouped together and given a single alternate text description. To verify image grouping, view the slide content in the *Navigator* dialog box (*Ctrl + Shift + F5*) and count that the number of the images shown to be on the slide is as expected.
3. Verify table row and if applicable column headers are provided.
 - a. For text documents, move the cursor within the bound of the table and select *Table > Properties*. On the *Rows and Columns* tab, verify the *Column header* and/or *Row label* checkboxes are checked.
 - b. Verify the number of rows and columns used as headers is not greater than one because it is difficult for screen reader users to interpret multiple row and multiple column headers.
 - c. For Symphony presentations, setting column header and row labels is not an available option. However, the user can use screen reader table commands to navigate the table and hear cell identification and content.
4. Verify the reading order and document navigation. The verification techniques are specific to each type of the Lotus Symphony content (document, presentation, and spreadsheet).

5. Run Symphony's spell check utility. (*Tools > Spell Check...* or F7.) Screen readers rely on correct spelling for correct pronunciation.

Verify reading order and document navigation for Symphony Documents

To verify the reading order, check whether there are complex frames or sidebars present; without them the document should read in the sequential order in which it is displayed. To verify navigation, check that all text used as section headings is formatted with the proper heading markup. Under the *Headings* node in the *Navigator* dialog box (*Ctrl + Shift + F5*), the text formatted as headings within the document is listed. Verify a table of content exists for long documents; it is recommended that a long document be quantified as one that is greater than 5 pages.

Verify reading order and document navigation for Symphony Presentations

To verify navigation, ensure the *Outline View* is populated with, at minimum, a uniquely named slide title. The outline view can be selected by the menu *View > Page > Outline view* or by using the view drop-down selector in the lower right of the page viewer. If two or more slides need to have the same title, add "cont." or "1 of N" to the title. The *Outline View* shows an organizational summary of presentation text. Pages created from standard layout templates will have their title and content automatically populated in the outline view. Text boxes and other objects added to a titled page will not display in the outline. Titles are important navigational clues to slides, and although other content will read with the slide, it is difficult to find a slide without a proper title. The outline view can also be selected using the view drop-down selector in the lower right of the page viewer.

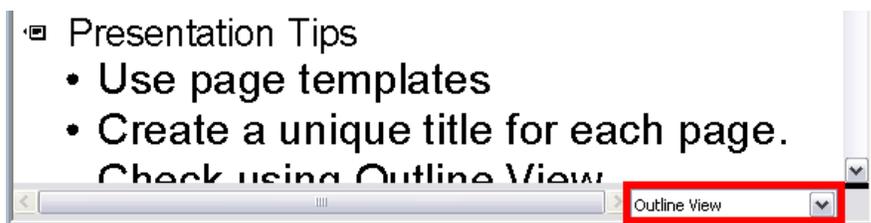


Figure 9. *Outline View* mode with the view drop-down selector highlighted

To verify the reading order of a Symphony Presentation, view the slides with the *Navigator* dialog box (*Ctrl + Shift + F5*) as described earlier in this paper. The screen reader reads Symphony objects in the order shown in the *Navigator*. Assure the order will make sense when using a screen reader. If not, adjust the reading order using the *Arrange* context menu item.

Verify reading order and document navigation for Symphony Spreadsheets

Verify the reading order by viewing the data to ensure that it is densely packed. No large blanks should be present; blank areas, blank lines, and blank columns should all be avoided.

Range names provide easy navigation. To verify navigation, use the *Navigator* command (*Ctrl + Shift + F5*) to ensure range names have been defined for the spreadsheet. Make sure there is a range name in the document called "information" or "instructions" with information on how to use the spreadsheet. Also make sure there is a final message on the last row such as "End of worksheet" to let the user know there is no more data.

Test with a screen reader

As discussed, most accessibility testing can be conducted using manual tests, but complex and high use documents can benefit from testing with a screen reader. The following steps help to ensure a successful test of Lotus Symphony content with a screen reader.

Ensure a valid test by restarting the system and starting the screen reader before Lotus Symphony. Some screen readers may fail to read Lotus Symphony content when started second.

Check a Symphony document with a screen reader by doing the following:

1. Read through the document with the screen reader.
2. Listen for logical order.
3. When links are used, verify the link text is meaningful and distinguishable from other links so it is useful when read by the screen reader.
4. Listen to the alternate text of the images and objects to ensure that the text properly describes the images and objects.
5. Ensure that the screen reader announces lists, tables, headings and other structures.
6. Verify that the column and any row headers of the table are spoken.
7. If the document is very long, additionally do the following: Listen to the entire table of contents.
8. Use the screen reader functions to list headers and links to assure they make sense when read in a list.

Check a Symphony presentation with a screen reader by doing the following:

1. Test the presentations in *Play Screen Show* mode. Press *F5* to start the screen show. Press *Page Up* and *Page Down* to advance the pages.

2. On each page, ensure all elements on the page are read in the correct order. Use the keyboard arrow keys to read individual elements one by one in order.
3. To read table rows and columns, use *Ctrl + Alt + (arrow keys)*.
4. If elements do not read in the correct order, adjust them using the *Arrange* context menu item. Screen readers read Symphony objects back layer to front layer.

Check a Symphony spreadsheet with a screen reader by doing the following:

1. Listen to headers and formatted numbers to make sure they read well.
2. Adjust text to read more coherently, where possible.

Summary / Conclusion

Lotus Symphony represents a complete, robust, and free accessible solution for creating open format text documents, presentations, and spreadsheets. Symphony is accessible to use, and its native document output as well as export formats to PDF and Microsoft Office can be made accessible in most situations. Testing Symphony content for accessibility is straightforward and similar in many ways to testing other documentation formats. Automated tools are becoming available to test ODF files. Testing in phases of increasing effort and greater detail helps find errors earlier in the process and reduces effort during accessibility testing.

References

Lotus Symphony is available free on the [Lotus Symphony website](#). Several [Lotus Symphony tutorials](#) are available to help the new user. Bugs and feature requests may be posted to the [Symphony support forums](#).

[“Introduction to OpenDocument Format”](#) Ali Sobhi, Senior Consultant, IBM, August 2006

Trademarks

- IBM, the IBM logo, Lotus, SmartSuite, and Symphony are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both.
- Adobe is a registered trademark of Adobe Systems Incorporated in the United States, and/or other countries.
- Eclipse is a trademark of Eclipse Foundation, Inc.
- Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
- Linux is a registered trademark of Linus Torvalds in the United States, other

countries, or both.

- Microsoft, Active Accessibility, Word, PowerPoint, and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.
- Other company, product, and service names may be trademark or service marks of others.

About the authors

Susann Keohane is a consultant with the IBM Human Ability & Accessibility Center. Her focus is on the accessibility of documentation and Java applications throughout IBM, with a primary goal of providing guidance to the Learning and Software Group Information Management teams. She is an IBM Master Inventor and holds a Master of Software Engineering degree from the University of Texas at Austin.

Brian Cragun is a consultant with the IBM Human Ability & Accessibility Center. He supports Lotus desktop products and testing applications. Brian is an IBM Master Inventor and has a Bachelor of Science in Computer Science from Utah State University and a Master of Science in Manufacturing Systems Engineering from University of Wisconsin - Madison.

Xing Li is an Advisory Software Engineer at IBM China, and the Chief Programmer of IBM Lotus Symphony team. He has rich experience on the development of Lotus Symphony. He has a Master of Computer Science from Tsinghua University of China.