Open Document Format in practice

WHO'S AFRAID OF ODT?

What happens when you feed an ODT document created with OpenOffice to a word processor like AbiWord, KWord, or Writely? Read on to find out. **BY KRISTIAN KISSLING**

Finally, we have an open standard for the exchange of data between office documents. The OpenDocument Format (ODF) is based on XML. You can use it to transport a text document (ODT), a drawing (ODG), a presentation (ODP), or a diagram (ODC) from OpenOffice to KOffice without any loss – at least in theory. Alternative word processors have not said much about OpenDocument Format thus far, and the implementation of this major standard is still in progress. Portability is a goal of ODT, and we wanted to learn how portable an ODT text document really is. To find out, we saved an ODT file in OpenOffice then opened it with tools such as AbiWord, KWord, and Google Docs.

A Short History of ODF

The ODF standard was developed by a technical committee known as OASIS [1] (Organization for the Advancement of Structured Information Standards). Although OASIS used the OpenOffice XML format as a template for the specification, more than 100 changes – and new capabilities – were added and the format went through extensive testing. OASIS experts investigated the specification for a whole year, which was followed by a one-month official test phase, and finally all OASIS members – representing some

600 organizations – voted on the standard. The OpenDocument 1.0 standard is thus based on a fairly wide consensus, and it looks likely to receive international approval, in the form of ISO/IEC 26300, in the near future.

ODF in Production Use

Our reference document was created with a OpenOffice Writer 2.0.2. It includes a few traps for the candidates (see the box "The Test Document"). The appearance of the final document is sensitive to details of the language, application settings, and version number.

OpenOffice Writer was the reference program because the developers used the Open Document Standard for OpenOffice, and the suite has more in the line of ODF features than other free office packages.

We opened a document created using the ODF specification in OpenOffice 2.0.1 and KOffice 1.5.2. It became apparent that neither of the two programs completely fulfills the specification [2], although OpenOffice supports most functions better than KOffice.

Implementations

For the test, we opened the text document with AbiWord [3], KWord [4], and Google Writely, which is now called Google Docs [5]. The latter is included because online editors increasingly support ODF. The advantage of an online editor is that you can work on a document via the Internet, no matter where you are. To use the online word processor, you need a computer with Internet access and a browser; avoid this approach for confidential data.

We opened the OpenOffice Writer reference document in the test candidates, compared the results before saving the document in the candidate's own ODT format, and then opened it again in OpenOffice to see the changes.

AbiWord

We were interested to see how the editors tackled this tough nut. AbiWord was the first to rise to the challenge; version 2.4.5 has just been released as an autopackage, and the changelog speaks of a number of bugfixes in the ODF support area. Irrespective of the alleged changes, the software skillfully dissected the test document to reveal its component parts (Figure 1).

The header still exists, and the font and font size are correct. However, the header is stuck at the top edge of the document. The frame with the title follows: AbiWord selected Times New Roman – although the Bitstream Vera Serif font is available – and simply adds a gray background to the letters. The remainder of the background in the frame is white. The image is on the right page, however, it overlaps the frame. The body text is hidden behind the image, and there is no gap between the text and the image. In other words, these three elements – the frame, the image, and the body text – all overlap, and the latter has moved upward.

The words in bold print in the body text survived the conversion, as have the footnotes, but the separating line is missing completely. The column text has bitten the dust; a break in the left column leaves the column completely empty. The text follows the break in the correct font and justified, as do two correctly formatted tables. However, there are three blocks of whitespace in the text where AbiWord has used a white font on white background, which might be useful for steganographic purposes, but is difficult to read. The footnote texts are glued onto the corresponding footnote number; the page number in the footer line might disappear when you try to print the document because it is close to the edge of the printable area.

Printing

As for printing, the hard copy again looks different. Despite the on-screen

The Test Document

The reference document includes various fonts, colors, columns, line spacings, tables, footnotes, and even an image. The header contains the title of the document in italics, centered, using the Century Schoolbook L font in nine point. This is followed by a gray box with a one-point frame. The frame contains the Title in 40 point, centered, and Bitstream Vera Serif.

The body text that follows uses the same font: it uses line spacing of 1.5, includes three footnotes, and includes four words in bold type. The text is justified, with a GIF image embedded on the right, with a slight gap to the left and below the image to allow the text to flow around it.

A separating line follows. Underneath this there is a three-column layout that includes text and tables. The italic, justified column header uses a 12-point Arial font. It includes three keywords that use white on a black background. The third column has two tables with a colored background. The test document includes three footnotes, and a footer line with the page number at its center.

display, there are pixel-sized gaps and overlaps between the cells in the tables. In the box, the body text now appears on the background instead of the white area, and the printer now randomly replaces the image with either a black or a white square.

If you now save the document in ODT format under AbiWord, and then open it up again in OpenOffice, you're in for another surprise – you are now looking at a completely new version of the document (Figure 2).

The image is missing. The body text has the wrong line spacing, but it is at the right position (below the frame). The frame now has a light-blue background instead of the gray background.

The words in bold type and the footnotes are somehow okay, and OpenOffice uses Times New Roman just like the template. The columns do still exist, however, now with single line spacing.

All that's now remaining of the original tables is the content; the original colors and lines have disappeared completely, along with the page number in the footer line. In the end, this turns out to be a fairly extensive mutation.

KWord

KWord is the KOffice word processor, which has supported the ODF standard for quite a while now. Again, we discover that the program has its own special interpretation of the ODT document (Figure 3), although the whole docum-nent ends up being far more readable than in AbiWord, because the elements do not overlap.

KWord uses the right fonts and justification, more or less. Everything is fine in the header line, the title, and the body text, although KWord replaces the Arial font with another font in the body text. Have you ever heard of a font called *AR PL ShanHeiSun Uni*? Surprisingly, KWord does not have the Arial font, and thus attempts to exchange it for an adequate replacement.

The background color in the box is also fine, but the software then trips over its own feet, placing the image on the left instead of the image on the right.

The body text no longer flows elegantly around the image but below it, and without leaving a gap between the text and the image.

The three words in bold type are still in the body text, and the footnotes are still hanging on in there. Oddly enough, KWord starts with footnote 0.

There is a separating line below the body text, but the program superfluously drops a black line on top of the footer area. The second line in the footnote is indented in KWord, a fact that might displace the layout if worst came to worst. The page number in the footer is fine, however, the blank to the right of the number is now missing.

Lowering the suspension on the body text ends up pushing the rest of the document down, and thus onto page two. The column layout does not actually contain any columns, but to make up for this, KWord now shows the inverted words correctly.

Finally, there are two tables with the right colors, but as in AbiWord, there are pixel-sized gaps between the cells.

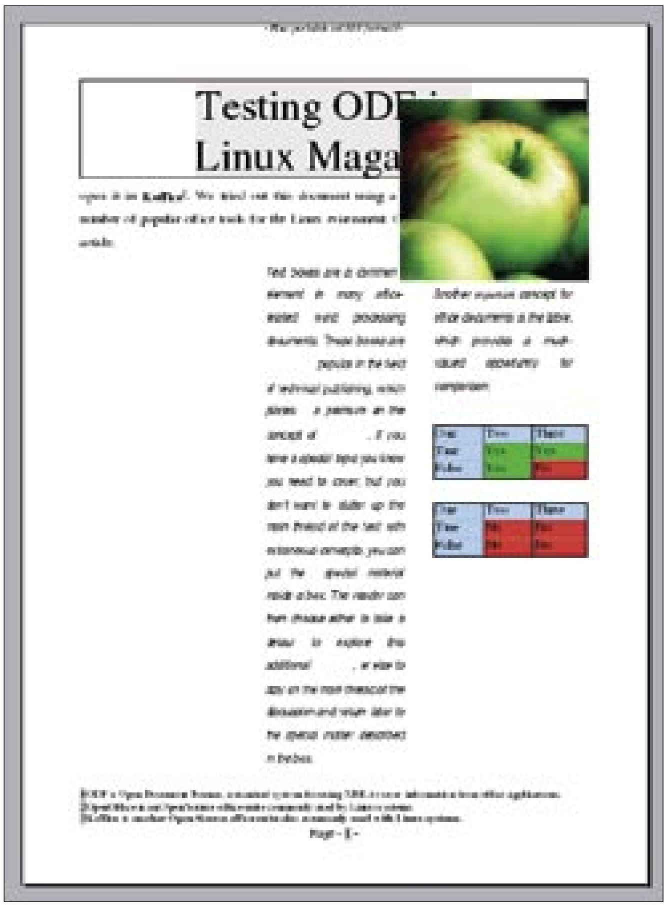

Testing ODF in Linux Maga

Figure 1: AbiWord in action: the similarity with the original isn't too convincing, especially with respect to the positioning of the elements and the colors.

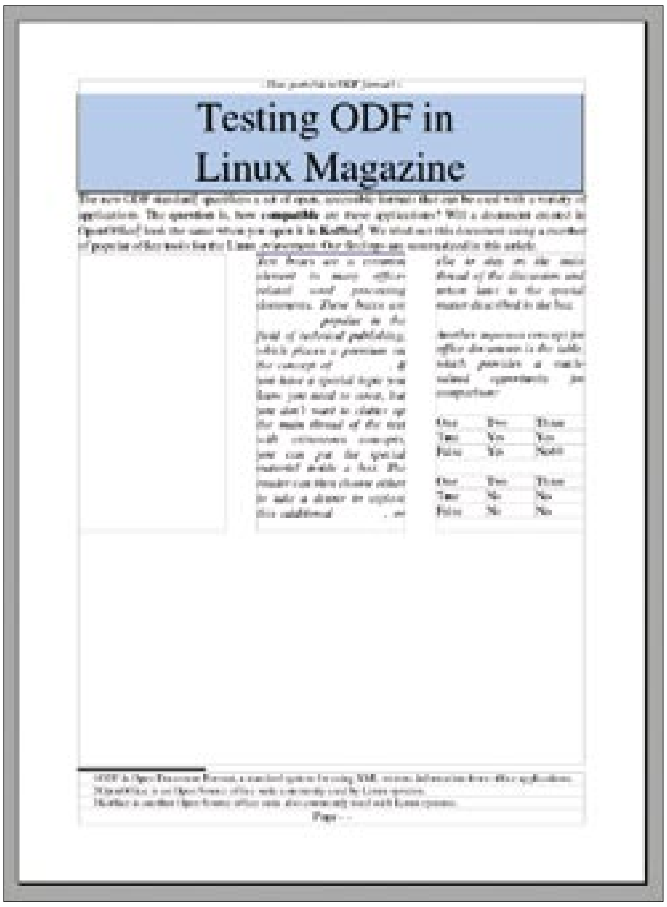

Testing ODF in Linux Magazine

Figure 2: Presto! If you save the document in AbiWord and open in OpenOffice again, you end up with something resembling a completely new document.

KWord obviously has its biggest problems when asked to handle a combination of text and images.

Hard Copy

The print-out looks a lot friendlier – you could even call it WYSIWYG. What you see is what your printer prints, however, the gaps between the cells in the table do look different on screen than in the hard copy.

If you then save the document in ODT with KWord, and open it up again in OpenOffice, you will end up with a completely new layout. The whole document now comprises three pages, the first of which is empty.

The image and the box with the title are now missing completely, with the second page bravely holding its own on page two. This time, we discover that the footnotes do start at the number 1 rather than 0.

On page three, the former three-column layout has still not recovered its columns. To compensate for this shortcoming, the tables now take up the whole width of the page.

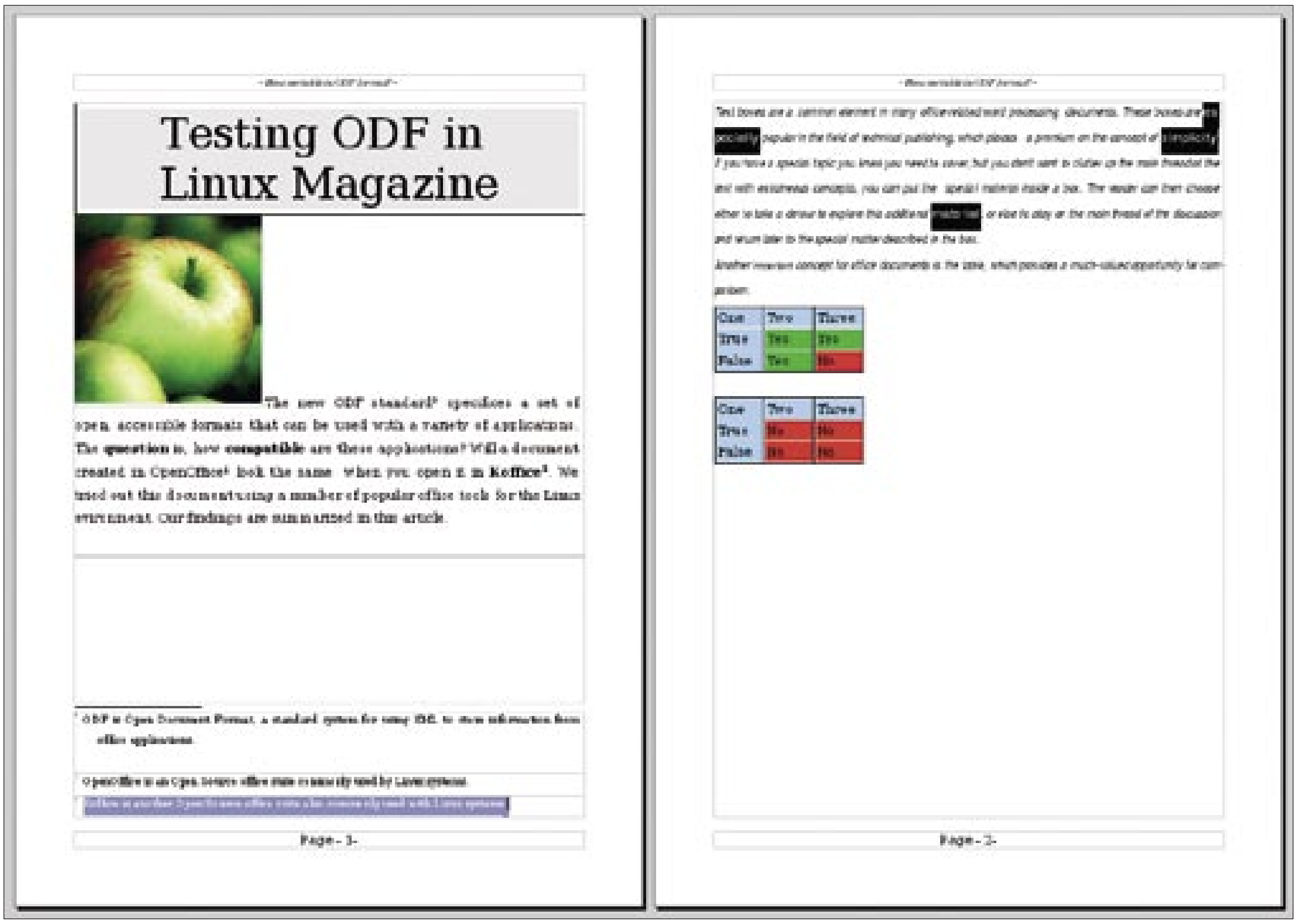
Table 1: Supported ODT Features

	KWord 1.6 beta 1	AbiWord 2.4.5	Writely
Headers and footers	●●●○○	●●●●○	●●●●○
Footnotes	●●●●○	●●●○○	●●●○○
Columns	●○○○○	●●●○○	●○○○○
Layers and element positions	●●●○○	●○○○○	●●○○○
Frames and separating lines	●●●●●	●○○○○	●●●●○
Text flows around image	●○○○○	●○○○○	●●●○○
Font style	●●●●●	●●●●●	●●●●●
Font type	●●●●○	●●●○○	●●●●○
Font size	●●●●●	●●●●●	●●○○○
Line spacing	●●●●●	●●●●●	●●●○○
Text color	●●●●●	●●●●●	●●●●●
Background color	●●●●●	●●●○○	●●●●○
Tables	●●●○○	●●●●○	●●●●○
Total (percent)	**45%**	**54%**	**52%**

Evaluation: total mark as a percentage

Writely

That just leaves us with the online candidate, Writely, which is now known as Google Docs. Will it be better equipped to cope with ODT? Well, not really, but the online app is not outpaced by its competitors, either. It is difficult to compare Writely directly with the other candidates, however; the software did not



Testing ODF in Linux Magazine

The new ODF standard[1] specifies a set of open, accessible formats that can be used with a variety of applications. The **question** is, how **compatible** are these applications? Will a document created in OpenOffice[2] look the same when you open it in **KOffice**[3]. We tried out this document using a number of popular office tools for the Linux environment. Our findings are summarized in this article.

[1] ODF is Open Document Format, a standard [illegible] to store information from office applications.

[2] OpenOffice is an Open Source office suite commonly used by Linux systems.

[3] [illegible]

Page - 3-

Text boxes are a common element in many office-related word processing documents. These boxes are especially popular in the field of technical publishing, which places a premium on the concept of simplicity. If you have a special topic you know you need to cover, but you don't want to clutter up the main thread of the text with extraneous concepts, you can put the special material inside a box. The reader can then choose either to take a detour to explore the additional material, or else to stay on the main thread of the discussion and return later to the special matter described in the box.

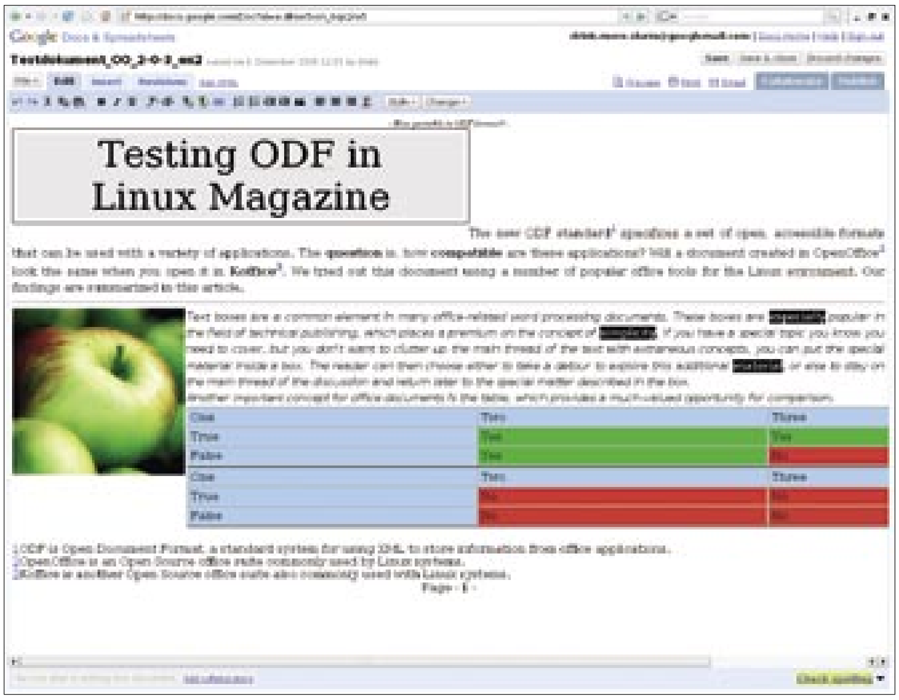
Another important concept for office documents is the table, which provides a much-valued opportunity for comparison.

One	Two	Three
True	Yes	Yes
False	Yes	No

One	Two	Three
True	No	No
False	No	No

Page - 2-

Figure 3: KWord has also supported ODF for some time now, although it does not support all the available options, as the test shows.



Testdokument_OO_2-0-3_ea2

Testing ODF in Linux Magazine

The new ODF standard[1] specifies a set of open, accessible formats that can be used with a variety of applications. The **question** is, how **compatible** are these applications? Will a document created in OpenOffice[2] look the same when you open it in **Koffice**[3]. We tried out this document using a number of popular office tools for the Linux environment. Our findings are summarized in this article.

Text boxes are a common element in many office-related word processing documents. These boxes are especially popular in the field of technical publishing, which places a premium on the concept of simplicity. If you have a special topic you know you need to cover, but you don't want to clutter up the main thread of the text with extraneous concepts, you can put the special material inside a box. The reader can then choose either to take a detour to explore this additional material, or else to stay on the main thread of the discussion and return later to the special matter described in the box.
Another important concept for office documents is the table, which provides a multi-valued opportunity for comparison.

One	Two	Three
True	Yes	Yes
False	Yes	No
One	Two	Three
True	No	No
False	No	No

1 ODF is Open Document Format, a standard system for using XML to store information from office applications.
2 OpenOffice is an Open Source office suite commonly used by Linux systems.
3 Koffice is another Open Source office suite also commonly used with Linux systems.
Page - 1 -

Figure 4: Writely is no worse than the offline contenders, but it displays the elements as an HTML page rather than a text document.

display the reference document in an A4 frame, but rather distributed it over the whole width of the screen. Because the office software resides on the server, it is difficult to know whether the application might be changing over time, but when I first tried to open the document, I got a result similar to the recreation shown in Figure 4. When asked to print the results, Writely squashed the whole of the document onto a single page.

Writely displayed the header correctly, although it changed the font from nine to 10 point. The program drew a narrow frame, changing the text background to gray and leaving the rest of the background white. As a replacement for Vera Bitstream Serif, which it does not support, Writely selected Times New Roman, but also new font sizes.

The title went to 36 point; the body text, which starts at the foot of the frame, went to 10 point. The bold words and footnotes looked fine. The line spacing was okay, although you could no longer change it, and the separating line was still there, although the column text flowed around the image rather than the body text (the former column text, as the columns had disappeared). The text flowed to the right of the image – a mirror image of the original.

The font and justification were fine, and the inverted words were still there. Writely showed the tables and their content as originally intended by the author, tagging them onto the body text, and using the correct background colors; however, the tables took up the whole width of the page.

That just left the footnotes. Each of our test candidates had its own, novel approach to footnotes. Writely paints them blue and underlines the footnotes in legacy HTML style. Just like in the original, the page number in the footer completed the document.

I tried to save the document in ODT format using Writely and again open it in OpenOffice. The columns were back and the table was in the right column. The column text still used single line spacing. So it did not fill out the page and the space between the two tables was missing.

Chinese Whispers

You may know the old party game for kids, Chinese Whispers. Everyone stands in a circle, and you whisper into your neighbor's ear; your neighbor passes the message to his neighbor, and on it continues until you return to the beginning. The last person in the chain says the sentence out loud; normally this is quite funny because the sentence has changed completely. Well, what you currently get with ODF is similar.

Although the developers are working toward implementing ODF, the marketers have promised too much. There is a world of difference between supporting and mastering ODF. None of the applications gave us the original document as the author intended. If you print the document, the layout changes; and if you reimport it, you might think it was a completely new document.

The good news is that you can open documents; you don't lose the text, and ODF support for simple documents is available. Ideally, the standardized format should give you an exact copy of the original, which didn't happen with any of the free programs – not even OpenOffice. Although OpenOffice was not one of our candidates, but our reference program, it still has some weaknesses.

What is causing the issues? First, the developers attempting to integrate ODF support are all on a different schedule. AbiWord seems to have started with columns, whereas Writely and KWord still ignore this feature. The editors have a bigger problem with fonts: if AbiWord does not have the right font, it will change your layout, after replacing the font. None of the programs can offer perfect handling of images, columns, and positioning of individual layout areas.

In the wake of the group that successfully developed the ODF standard, it seems we now need a second group to ensure standardized implementation. Inge Wallin, the Marketing Coordinator with KOffice, summarized the situation in a blog comment: Full support for ODF is a long-term target, and it is little surprise, if you take the scope of the standard into consideration, that the programs are incompatible in this phase. Unless you really enjoy a good game of Chinese Whispers, you might reconsider exchanging ODT documents. ■

INFO

[1] OASIS: *http://www.oasis-open.org/home/index.php*

[2] Akademy test of OpenOffice and KOffice: *http://netmoc.cpe.ucf.edu/Projects/OpenDocument/TestSuite.html*

[3] AbiWord: *http://www.abisource.com*

[4] KOffice with KWord: *http://www.koffice.org*

[5] Writely online editor: *http://www.writely.com*