

KOffice

A General Introduction

Raphael Langerhorst and Jost Schenck



KOffice

Contents

1	Introduction	1
1.1	KOffice components	1
1.2	Overview of KOffice features	2
1.2.1	Integration	2
1.2.2	Lightweight	2
1.2.3	Completeness	2
1.2.4	OASIS OpenDocument Format	2
1.2.5	KDE Features	3
2	KOffice Technology	4
2.1	KParts - the building blocks of KOffice	4
2.1.1	Compound documents with KParts	4
2.1.2	Using KParts in your document	5
3	Configuring KOffice and Your System	6
3.1	Optimizing Font Output	6
3.1.1	How to Get Nicely Scaled Fonts on Screen	6
3.1.2	How to Get Correct Printout	6
3.2	Customizing the KOfficeGUI	7
4	How to get more information	9
4.1	Other KOffice manuals	9
4.2	Links	9
5	Programming KOffice	10
5.1	Introduction	10

6	Copyright and Licensing	11
A	Installation	12
A.1	System Requirements	12
A.2	Getting Binary Packages	13
A.3	From Binary Packages	13
A.4	Getting Source Code	14
A.5	From Source Code	14

Abstract

KOffice is an integrated office suite for the K Desktop Environment (KDE).

Chapter 1

Introduction

1.1 KOffice components

KOffice is an integrated office suite for the K Desktop Environment (KDE). KOffice currently consists of the following components:

- KWord (a frames-based wordprocessor)
- KSpread (a spreadsheet application)
- KPresenter (screen and slide presentations)
- Kivio (a flowchart application)
- Karbon14 (a vector drawing application)
- Krita (a pixel based drawing application)
- Kugar (a tool for generating business quality reports)
- Kexi (an integrated environment for managing data)
- KChart (a charts/graphs generation application)
- KFormula (an editor for mathematical formulae)

Because these components are based on the KParts component model, KOffice components are designed to work very well with each other. Any KOffice component can be embedded in another KOffice document. For instance, you can insert a spreadsheet which you created in KSpread directly into a KWord document. In this way, complex, compound documents can be created using KOffice.

A plugin mechanism makes it easy to extend the functionality of KOffice. You will find many plugins in some of the components and can learn how to write

plugins yourself. There is also a section of this manual dedicated to developing plugins that should get you started.

This manual only covers the general features of KOffice, those that are common to most or all components in the suite. For detailed information about the different components, have a look at the respective manual.

1.2 Overview of KOffice features

1.2.1 Integration

KOffice provides a framework that all components build on. Through this framework high integration is achieved. It is possible to develop your own component that integrates as well. The technology of this is described in more depth in the [chapter on KParts](#).

1.2.2 Lightweight

Despite offering so many components, KOffice is very lightweight, fast and has a rather low memory footprint. This is also achieved because KOffice builds on KDE, which is already a very powerful application development framework.

A goal of KOffice is also to not overload the user interface with features that are hardly needed. This results in a lean and mean user interface that lets you do your work efficiently.

Being lightweight can at times also mean that that very particular special functionality you need is not there. In this case you can always add the functionality yourself by extending KOffice. KOffice really doesn't want to get bloated by adding large chunks of features that only very few users might need. Such features can always be made available through additional plugins or scripts, thus keeping KOffice itself lightweight.

1.2.3 Completeness

KOffice offers a wide range of components, covering most of the home and business office needs. Additional features can always be implemented through scripts, plugins or even whole components based on the KOffice framework.

1.2.4 OASIS OpenDocument Format

It is a major importance for any office suite to adhere to established standards. Especially on the file format level to allow seamless document exchange with other office suites. This also avoids vendor lock-in, which is especially important for companies and also for individuals.

KOffice

For this reason KOffice has adapted the OASIS OpenDocument format as native file format.

1.2.5 KDE Features

Since KOffice builds on KDE all the features you would expect from a KDE application are also available in KOffice. This includes DCOP, KParts, Desktop Integration, Configurability and so on. All this makes the look and feel of KOffice very familiar and really integrates KOffice into the desktop environment, thus allowing seamless workflows.

Chapter 2

KOffice Technology

2.1 KParts - the building blocks of KOffice

Each KOffice application is designed to fulfill very specific needs. For example, KSpread is a program for manipulating spreadsheets, while KWord is a program for word processing. By focusing on a specific area, each program aims for perfection in exactly this area. However, depending on what you do with KOffice, you'll often want to take advantage of functionality provided by different applications, but in the *same* document.

Suppose that you are preparing a document in KWord and want to illustrate some point using a table. While KWord provides its own functionality for the insertion of tables, this may not be enough for your needs. For example, you may want to use certain currency data formats or to do calculations using spreadsheet formulae. Now, some KWord programmer certainly *could* try to implement this functionality. However, KWord would never be as good as KSpread in this field; and if it tried to implement all the functionality you might possibly need, it would end up becoming unreasonably complex and the source code would become impossible to maintain.

The alternative to this is KParts. Its philosophy is simple: let every application do what it does best.

2.1.1 Compound documents with KParts

With KParts, your documents can use *all* functionality offered by *all* KOffice applications. You can take advantage of this by inserting so-called 'parts' into your document. Every one of those parts is essentially another document, that is, 'a document within a document'.

In the example mentioned above, you would simply insert a KSpread part into your KWord document. Then, whenever you edit your table, KSpread would

take control in the background. Control would return to KWord when you stop working on the table and start working on text again. The only change you would notice is that the toolbars and menus, while editing the table, reflect the functionality of KSpread instead of that of KWord. This ability to include the functionality of one application in another is called 'embedding'.

2.1.2 Using KParts in your document

If you have never worked with compound documents, you may find them confusing at first. The following step-by-step instructions show you that using KParts is just as easy as working with a single application.

1. Start KWord. You can do that either from the panel or by typing **kword** at the command line.
2. Start a new, blank document. You may want to type some sample text.
3. Select Insert → Object Frame → Spreadsheets. You'll notice that the mouse cursor has changed to a cross. This form of the cursor indicates that you are supposed to select an area.
4. Press the left mouse button where you want the upper left corner of your table to be, hold it and drag it to the lower right corner. Release the button. Now a dialog appears that gives you the possibility to insert an existing document or to create a new one. Create a blank worksheet. That's it--you're done.

Easy, isn't it? You should now see a table frame in your KWord document. Now, doubleclick inside the table to see what happens. You'll notice that:

- Your menubars and toolbars have changed. They now reflect those of KSpread.
- Your table frame now contains the elements of a KSpread view, with scroll-bars, a tab bar for the selection of tables, etc.

Try editing the table. You'll see that it's not different from using KSpread. In fact you *are* using KSpread.

Now click somewhere into your KWord document, outside of the table area. The menubars and toolbars should change back to those of KWord, the elements of the KSpread view should disappear. Your table stays and still reflects all of the changes you've applied to it.

KParts are easy to use. Try inserting other parts or even parts in parts.

Chapter 3

Configuring KOffice and Your System

While KOffice should work quite nice out of the box, there may well be some things to optimize to get the best out of KOffice. This chapter shows you what you might want to do to achieve the best results with your new office suite and make it suite your needs. KOffice is highly configurable, even down to detailed toolbar and menu layout.

3.1 Optimizing Font Output

Fonts are a difficult topic on X Windows. In this section we'll cover some problems that are frequently reported by people using KOffice. Some problems are not just KOffices fault, but depend on your system configuration, which is why you may need to modify system configuration files in order to solve them. If you don't have access to the root account on your system, please ask your system operator about this and point him or her to this manual. As the topic of fonts is too complex to cover all of it here, you may want to consult the [Font HOWTO](#) from which I've taken the following information. You will find more details there.

3.1.1 How to Get Nicely Scaled Fonts on Screen

STUFF

3.1.2 How to Get Correct Printout

While KOffice automatically can handle all X11 fonts *on screen*, printout can pose a problem: on most systems, printing is done via ghostscript. Now, while

KOffice

KOffice knows the font names used by X Windows, it does normally *not* know the font names used by ghostscript. KOffice tries to guess these names, which unfortunately doesn't work all of the time.

This problem can be solved, although this is not that easy. Actually, maybe you are using a distribution which has done most work for you already (so if you have no reason to complain about printout you can skip this section). What you have to do is to tell ghostscript how to translate the (guessed) font names KOffice uses to its own font names. This can be done by adding lines to a file called `Fontmap`. An alias line in `Fontmap` looks like the following example:

Example 3.1 An alias in the ghostscript `Fontmap`

```
/Algerian-Roman /Algerian ;
```

Please note that a space before the `';` is mandatory. In this example, `Algerian-Roman` is the name KOffice uses for `Algerian`. You'll have to add such lines for the fonts KOffice doesn't display correctly. To make this task easier, Donovan Rebbechi has written a perl script you can find at http://pegasus.rutgers.edu/~elflord/font_howto/kwdfont. Assuming that you have a fontfile `/usr/share/ghostscript/fonts/fontfile.ttf` you'll enter `kwdfont /usr/share/ghostscript/fonts/fontfile.ttf` to get the appropriate aliases. The script should work in most cases. As mentioned, you should have a look at the [Font HOWTO](#) for more accurate and in-depth information.

3.2 Customizing the KOfficeGUI

While KOffice comes out of the box with a GUI (graphical user interface) that should suit most people's needs, there are good reasons why you may want to change the way the programs look.

My mother, for example, is a bit afraid of buttons and menu entries she doesn't understand. To tailor KOffice to her needs, I reduced the GUI to a bare minimum of functionality. As she only needs to write letters and use certain templates, there is no need for much more functionality than saving, loading, printing, etc.

Thanks to the 'action' concept of QtTM and KDE, you can freely customize KOffice menubars and tool bars. Unfortunately, at the moment, there are no easy-to-use dialogs to do this. KOffice stores its GUI configuration in XML files and you'll have to edit those. Hopefully, this situation will change in the future; for now, you'll need some basic knowledge of how an XML document works (or HTML, which is a subset of XML). [The 'action' concept needs to be discussed in more detail -- kt.]

Normally, each KOffice application has at least two of those XML files: one describing the GUI of the shell (basically, that's what you see if there is no document open) and one describing the GUI of the respective part (that's what

KOffice

you see normally). For example, for KWord, these two XML files are named `kword_shell.rc` and `kword.rc`.

Here's a simple example of such an rc-file.

Example 3.2 An example of a simple rc-file

```
<!DOCTYPE QConfig ><qconfig>
<menubar>
<menu name="Edit"><text>Edit</text>
<action name="edit_cut"/>
<action name="edit_copy"/>
<action name="edit_paste"/>
<separator/>
<action name="edit_find"/>
</menu>
<menu name="Insert"><text>Insert</text>
<action name="insert_picture"/>
<action name="insert_clipart"/>
<menu name="Variable"><text>Variable</text>
<action name="insert_var_datefix"/>
<action name="insert_var_datevar"/>
</menu>
</menu>
</menubar>
<toolbars>
<toolbar name="edit_toolbar" position="right">
<action name="edit_cut"/>
<action name="edit_copy"/>
<action name="edit_paste"/>
</toolbar>
</toolbars>
</qconfig>
```

Chapter 4

How to get more information

4.1 Other KOffice manuals

For detailed information on the different KOffice applications, please consult their respective manuals.

4.2 Links

The following links should be useful if you're looking for more information about KDE or KOffice.

- [The KOffice homepage](#). Here you can find information on how to get and install KOffice, news about KOffice development, screenshots etc.
- [The KDE Homepage](#). KDE is the most advanced and absolutely free desktop environment for unix-like operating systems. KOffice makes use of the KDE libraries.
- [Trolltech](#). The creators of the C++-toolkit Qt™. KDE and KOffice make use of Qt™.

Chapter 5

Programming KOffice

5.1 Introduction

If you want to do KOffice development, the following resources might be of interest:

- At <http://developer.kde.org> you'll find many documents about programming with Qt™ and KDE. Here you can find the online version of the complete KDE library documentation.
- In the KOffice sources you'll find an example application in the `example` folder.

Chapter 6

Copyright and Licensing

KOffice is the result of the joint effort of many developers. Each source file of KOffice is copyrighted by the people who wrote this particular file and their names can be found at the start of each source file with the license that applies to that source file. The names of the core developers can be found at <http://www.koffice.org/people.php>.

This manual is copyright by Jost Schenck. It can be distributed freely, as long as this copyright notice is included. You may change it as long as you send me the changes or commit them to KDE CVS. I'll not be liable for anything resulting from the use of this manual.

The other KOffice manuals are copyrighted by their respective authors.

This documentation is licensed under the terms of the [GNU Free Documentation License](#).

This program is licensed under the terms of the [GNU General Public License](#).

Appendix A

Installation

A.1 System Requirements

In order to install and use KOffice you must have:

- A functioning UNIX® system (for example, Linux® or BSD) with a configured X Window System® (for example, XFree86 or X.Org).
- The Qt™ 3.3 libraries from Trolltech. See www.trolltech.com for more information.
- The K Desktop Environment 3.3 (KDE 3.3) or newer. KOffice cannot be compiled with earlier versions of KDE. Information on obtaining and installing KDE can be found at: <http://www.kde.org/>
- Permission to install software on your computer. If you are on a stand-alone system, this should not be a problem. However, if you are using a networked computer, check with your administrator first.

Please note that, while you need the KDE 3.3 (or newer) libraries on your system, you can still use KOffice in a different desktop environment (for example, XFCE or GNOME).

If you plan on [compiling KOffice from source code](#) you will also need:

- **automake** 1.6.1 or later. This can be downloaded from: <ftp://ftp.gnu.org/pub/gnu/automake/>
- **autoconf** 2.53 or later. This can be downloaded from: <ftp://ftp.gnu.org/pub/gnu/autoconf/>
- A C++ compiler which supports exceptions, preferably a recent version of GCC. (See <http://gcc.gnu.org> for info about getting and installing it.)

KOffice

- Troll Tech's Qt™ 3.3 development packages. If these packages are not installed on your system and they are not included in the distribution of your operating system, you should probably have to compile Qt™ from source. These sources can be obtained from <http://www.trolltech.com/download/-index.html>.

Once you have determined that your system meets these requirements, you should decide whether you want to compile the source code or install from pre-compiled binary packages.

If you want to obtain the most current binary version of KOffice, you can find more information by following this link: [Getting Binary Packages](#).

If pre-compiled binaries are not available for your system, you can download the source code and compile it yourself. Instructions on where to obtain the current source code (and what to do with the source code once you have it) can be found by following this link: [Getting the source code](#).

NOTE

Please see the [KOffice homepage](#) for further details. That is where you can always find the most up-to-date information on both source and binary distributions!

A.2 Getting Binary Packages

You can download the most current binaries from:

<http://koffice.kde.org/releases/1.4.0-release.php>

or from one of many mirror sites. The current list of mirror sites can be found at:

<http://www.kde.org/mirrors.html>

NOTE

It is important that you use a binary package which is compatible with your system. If you use the Linux® operating system and are unable to find a binary package at the KOffice web site or one of its mirrors, you may be able to obtain one from the website of your distribution.

A.3 From Binary Packages

To install KOffice from precompiled binaries, you should:

1. Ensure your computer has all of the required software installed and in working condition (excluding KOffice of course). You can find the system requirements by following this link: [System Requirements](#).

KOffice

2. Download the KOffice binary (or binaries) into a temporary folder.
3. If the filename ends in `.rpm` (Red Hat® Package Management file), KOffice can be installed with the following command:

```
$rpm-U filename
```

If the filename ends in `.deb` (Debian Package file), KOffice can be installed with the following command:

```
$ dpkg-i filename
```

If the filename ends in `.tar.gz` or `.tgz` (a tarball file), KOffice can be installed with the following commands:

```
$ cd /  
$ tar -xzvf filename
```

In these examples *filename* should be replaced by the complete name of the package *including the full path* if you are not in the directory in which you saved.

4. That's all. KOffice should now be installed on your system.

NOTE

If you have a graphical front-end for package management, such as KPackage or GnoRPM, installed on your system, you may find that more convenient to use than a command line. Consult the program's documentation to find out how to install with it.

A.4 Getting Source Code

You can download the current source code from: <http://koffice.kde.org/> or from one of many mirror sites. The current list of mirror sites can be found at: <http://www.kde.org/mirrors.html>

A.5 From Source Code

If you want to build KOffice from source code, you should:

1. Ensure your computer has all the required software installed and in working condition (excluding KOffice of course). You can find the system requirements by following this link: [System Requirements](#).

KOffice

2. Download the KOffice source file into a temporary folder.
3. If the filename ends in `.src.rpm` (Red Hat® Package Management file), the source code for KOffice can be installed with:

```
$ rpm-U filename
```

If the filename ends in `.src.tar.gz` or `.src.tgz` (a tarball file), the source code for KOffice can be installed with:

```
$ tar -xzvf filename
```

4. The source code for KOffice should now be installed on your system.