



From: Manik Chand Patnaik,
C/O Gopal Sundar Patnaik,
SUNDAR NIVAS, Karan Street,
AT/PO: Paralakhemundi,
PIN Code: 761200,
State: Orissa,
INDIA.

Paralakhemundi/15-Feb-2006

To: koffice-competition@kde.org
The GUI and Functionality Design Competition for KOffice2.

Subject: Competition Entry.

Esteemed Judges,

This document is a humble submission for the GUI and Functionality Competition for KOffice2. It contains UI Ideas, Mock-ups and other Functionality issues along-with their Usability analysis. Pointers are provided to guard against probable pitfalls. I have tried to present the important points appropriately stressed out by the use of topic statistics with five star indicators indicating their priority and usability levels.

About Me: I am a citizen of India and belong to the State of Orissa. I hold Master Degrees in Finance (Master in Finance and Control) and also in Information Technology (M.Tech. – IT). Since a year or so, I am doing a functionality analysis of Open Source Applications. This document is built upon this experience of mine.

Analyzing and visually representing an office suite is a huge task. The trio of the Wordprocessor, Spreadsheet and Presentation software constitutes the bulk of any office suite. Though I have tried to amply analyze these three areas of operation, the Wordprocessor deserved special attention because of the importance associated with it. This representation constitutes discussions over many general issues which are applicable to all the three products under focus. The practical usability in the end user's point of view have been put forward with a higher priority level than the fancy features.

Keeping with the spirit of Open Source, all the User Interface Ideas and Usability analysis presented herewith can be shared under the GNU Free Documentation License. So these ideas can be freely applied for the upcoming releases of KOffice which is to be released under GPL.

Thanking you for providing the end-users with the opportunity of analyzing and presenting their suggestions about KOffice on such a massive scale.

Regards,

Manik Chand Patnaik



KOffice UI and Functionality Suggestions with Mockups and Feature Explanations

Author: Manik Chand Patnaik

Mail : myself -at- manik.in

Contents . . .

Introduction [↗](#)

General Topics

1. Platforms and Application Software [↗](#)
2. Organizing Documents with Meta-data (Type and Classification) [↗](#)
3. Font packaging with document (Font Embedding/Archive packaging/Subsetting) [↗](#)
4. Configurable/Skinnable GUI (Font Names, Colour Values or Graphics) [↗](#)
5. Keyboard Shortcuts [↗](#)
6. Wizards/Druids [↗](#)
7. Backing up of Program Settings [↗](#)

Specific GUI Topics

1. Start-Up Screen [↗](#)
2. Menus, Tool Bars and Shortcut Menu [↗](#)
3. Ruler, Guide, Tabs and Margins [↗](#)
4. Client Area [↗](#)
 1. Non Obtrusive Dynamic Worksheet [↗](#)
 2. Assisted Typing Technology [↗](#)
 3. Help Bubbles [↗](#)
 4. Mini Tool Bubbles [↗](#)
 5. Worksheet Rigidity [↗](#)
 6. Automatic Obstruction Removal [↗](#)
5. List Handling [↗](#)
6. Tables and Other Objects [↗](#)
7. Extra Context based Tools: Docks [↗](#)

Special Discussion on The Word Processor [↗](#)

1. Improving KWord as a documentation Tool [↗](#)
2. Content Creation and basic formatting [↗](#)
3. Page, Paragraph and Text Formatting [↗](#)



4. Multi-Level documents [↗](#)
5. Multi Layered documents (Master and Meta Documents) [↗](#)
6. Professional Pagination and Pre-Press work [↗](#)

Special Discussion on Spreadsheets [↗](#)

Special Discussion on Presentations [↗](#)

Document Security Encryption and Signing

1. Selectable Encryption Support [↗](#)
2. Digital Signing support (Digital Certificate and GPG) [↗](#)

Montage and Show-Packaging [↗](#)

Concluding Remarks [↗](#)



INTRODUCTION

About this write-up

This write-up aims at bringing together different concepts for an over-all improvement of any office suite and this includes KOffice. During the preparation of this article care has been taken to look in to the **practical usability** of the suggestion and **implementability** of a solution. Specific markers on the usability angle of the suggestion is specified by star rating. The suggestions are categorised into two sets. The first set of suggestions are general in nature and the second set of suggestions are very specific for particular elements of the program Interface or Behaviour.

For a better representation of the vitality of the suggestion a small list of vital statistics precede each individual article. It takes the form of the following example.

Type	General	//type of suggestion
About	Some description here	//basic information
Audience	Target Audience	//the category of people //who will use this feature
Usability	(★ ★ ★ ★ ★)	//Usability index (out of 5)
Necessity	Hi	//Urgent need for implementation
Implementability	(★ ★ ★ ★ ★) Easy/Hard	//How hard it is to implement

Special attention has been given to the Usability section in this vital statistics list. The Implementability may be easier or harder than it seems to me but I don't expect much deviation. The Necessity is expressed in words for a better representation. As Necessity is quite abstract in nature, a five star rating is not followed here.

[Back to Contents](#) 



Platforms and Application Software

Type	General
About	KOffice for multiple platforms
Usability	(★ ★)
Audience	All KOffice Users
Necessity	Hi
Implementability	(★ ★ ★ ★ ★) - Hard

Platforms and Application Software

The usability of an Open-Source desktop depends a lot on the applications available for it. For an average desktop user the availability of a good Office Suite is a recognized necessity. In this scenario there are several choices available for the desktop user. Most of the recognized Office-Suites from the established vendors in the market are targeted towards the Windows platform. This effectively limits the choices of the user adopting the free alternatives available. The two notable open-source and free office products are OpenOffice and KOffice. From these two the former is available on many platforms inclusive of Windows and MacOS. This way it has been able to garner support from a wide variety of users. As for now porting of the KDELibs has not been possible due to commercial license restrictions of QT3 and the announcement of TrollTech on the availability of the GPL version of QT4 for Windows has opened up many possibilities for porting of KDE applications across platforms.

I have seen in the past how the availability of Open-Source applications across platforms have contributed to the adoption of the FLOSS in general computing. Most users, serious about the switch have tested and satisfied themselves with the ports of many of the FLOSS applications in their native environment. Flagship FLOSS applications like FireFox, OpenOffice and Gimp have brought many enthusiasts into the folds of Open-Source. Now it is the turn of KDE and related technologies to achieve the kind of support and admiration on other platforms the above-stated applications have enjoyed. KOffice is still limited to the Unix/Linux side of the Operating Systems. Hopefully the next generation of KOffice breaks free from the platform shackles and Win friends across platforms.

[Back to Contents](#)



Organizing Documents with Meta-data (Type and Classification)

Type	General
About	Document Management
Usability	(★ ★ ★ ★ ★)
Audience	All KOffice Users
Necessity	Relatively High
Implementability	(★ ★) - Application Intelliscence is a bit hard

Documents and Meta-data

Generally a common Office Suite user creates and receives a variety of documents. These files (referred here as Office Files) slowly pile-up in the hard-disk and it becomes very difficult to track it down later. This situation is quite common everywhere. Efforts both from the user and the Application vendor is needed to remedy this situation. As computer files are not designed to be organised by default, specific measures need be taken to enhance organisability in office documents. The most common way to enable organisability is with the use of Meta-Data. Meta-data is some extra information about the document and meaningful and accessible ways should be provided to tempt users to enter Meta-data. Several ways to accomplish the task are explained here.

Extra Options in Save Dialog

The save dialog is a convenient spot for entering Meta-data. An extra button named Meta-data can be put there which would expand the save dialog itself rather than open a new dialog. The past behaviour with the dialog (if the user has been willing to add extra information) should take priority when the save dialog is shown next time (Normal/Expanded State). Even in the normal state a drop-down combo should be offered which would have several pre-set document definitions like Letter, Memo, Fax, Envelope etc. This would tempt the user to either type in the custom classification or accept one of the pre-set ones.

Automatic Categorisation during Creation

During the creation of the document the user generally builds upon an existing template.



The New Document dialog should provide templates targeting easy categorisation. For example templates named “Letter” “Fax” and “A4 document” may seem similar but the meta-data they automatically generate would help categorisation of the documents later.

These automatically diagnosed document categories should be fed into the save dialog for confirmation of the user during a save operation. If the user accepts/ignores it the categorisation will remain valid unless the user may enter his own choice in the save dialog.

Currently the situation is so bad that there is rarely any meta-data saved alongwith the document, leave aside advanced categorisation. OOo's Document information does not have a sheet for Author data. KOffice provides a good Document Information sheet. What's needed is ways to tempt the user to add information in it and tools in the Office Workspace to search on these extra information saved with the document.

Intelligent Categorisation

When the document is being created, special objects like addresses, images, legal addressing may be found in the document. The application can find these out through Application Intelliscense. This can result in some good categorisation of the documents.

Adding Meta-Data in Existing Documents

If “Intelligent Categorisation” is implemented, it can extract meaningful information from any document, not necessarily new ones. So saving the meta-data with the document makes sense. While closing an open document which has no meta-data, the user may be asked to review and save the extracted meta information. However this prompting can become quite nagging. So option to disable it temporarily for the current session should be added. A simple log-out and re-log-in would re-enable this behaviour. However to disable this behaviour, the user should change the application settings for it. Providing options to permanently disable this prompt would be disastrous because the user will rarely go to the settings to re-enable prompting.

The Office Workspace should provide options for organising office files. A user would feel free to add and modify meta-data in the existing documents during his leisure if he finds a preview based option in the Office Workspace.

As a sample: there would be a preview pane and a meta-data pane along-with a file selection pane. The user will select the document from the file pane, view the file in the preview pane and add the information in the meta-data pane.



Organising Documents using Folders

Using folders to organise files is being used from quite a long time. However software applications never provided support for this. Organising office files through the Office workspace should be easily achieved. Spreadsheets, Text Documents and Presentations should be placed in different folders. However many won't agree with me because it will add three more folders to the “Document folder” in your “home” but rest assured, it will improve manageability in the long run. By the way, as a side note: many small users won't be needing the Spreadsheets folder.

[Back to Contents](#) 



Font Packaging with Document (Font Embedding/Archive packaging/Subsetting)

Usability	(★ ★ ★ ★)
Audience	All Office Suite Users
Necessity	High (Not having this feature has been quite problematic till now)
Implementability	(★ ★ ★) - Moderately Hard (Complex Features)

The problems with fonts for editable and non-editable documents is not new. The solutions for them have been developed long back too. Even the current generation of Unicode fonts have flags to allow/restrict embedding. I am a font author and I find it embarrassing that no present generation word-processor is having any commitment towards the “Editable Embedding” and “Installable Embedding” of the fonts. This is a very sorry state. A replacement table can at most show the text in a different font but any-one with a sane understanding of page layout would know how a replacement font can screw-up the entire formatting. The current text utilises “Utopia” for running text, CMU Sans Serif for Chapter and Sub-Chapters. Opening this file from Windows where these fonts are not present makes this document look really horrible. Neither OpenOffice nor KOffice gives me an opportunity to bundle these fonts along the document. After-all OASIS file formats are archives and fonts with “installable embedding enabled” can be bundled together. However for editable-embedding and printable-embedding enabled fonts modification to the font is necessary. Further more editable and printable embedding expects the application to render fonts from the application itself. It is not impossible but it needs work (PDF files do embedding and subsetting of fonts for distribution for read only documents and QT4 series does create PDFs quite well but we need an editable solution for our Office Suites). For “Printable Embedding” subsetting of the font may be necessary to prevent implied distribution.

The solution to this problem is as complex as the problem itself. However somebody should take the initial step so that solutions can get implemented. This would be a boon for the people from the publishing industry and quite useful for general and corporate/enterprise users.

Suggestions for managing the Font problem:

- a.) Add the Installable font to the OASIS document as a font file.
- b.) Temporarily Auto-install the missing Installable font in the user's home for viewing and prompt to user to leave the font there for future use and if denied delete the font file.



- c.) Package the font with “Editable Embedding” with the document file as an encoded font either through some encoding like base64 or UUE and at the run-time unpack the font for use in a temporary location to be deleted during closing of the document.
- d.) Package the font with “Printable Embedding” with the document file as an encoded font after subsetting (Can Qt handle SVG fonts? They are easier to manipulate). This would make the file editable but not with the “current selection of font” and printable with the “current selection of font”.

[Back to Contents](#) 



Configurable/Skinnable GUI

(Font Names, Colour Values and Graphics)

Usability	(★)
Audience	ISVs
Necessity	High for the vendors
Implementability	(★) - Quite Easy (can be done with most FLOSS applications)

Configurability and Skinnability are two aspects of a program generally for tweekers and customised solution providers. This is about ways to make it modified/personalised. This is quite necessary for a packager/distributor's point of view. An easy way to skin the application through installation of a skin goes a long way for garnering support from independent distributors who want to customise the solutions for their specific customers. These values need not be exposed through the settings dialog but they should be exposed through configuration files. There should be a configuration guide/script for this purpose directed towards the specific release of the software. So that the program could be customised even after the initial installation.

For the Skinning the following bears significance:

- a.) Colours including the “Document Background Colour” should be changeable. The purpose:
 - 1.) Readability and reducing stress on eyes.
 - 2.) Accessibility for disabilities.
- b.) All the graphic elements (Icons and other Images) should be changeable (also should be able to reset to the default scheme).
- c.) Through configurations KOffice should be made to use a specific widget theme. (Although all KDE application can be asked to use a different style than what is specified in Control Centre.)
- d.) Fonts utilised for different GUI elements like Menus (Currently governed by the Desktop Environment Settings) should be changeable for KOffice only.

In short KOffice should seek independent existence as a stand-alone product.

[Back to Contents](#) 



Keyboard Shortcuts

Usability	(★ ★ ★ ★)
Audience	All Office Suite Users
Necessity	High
Implementability	(★ ★) - Relatively easy

After a user gets accustomed to his work environment, he needs performance enhancements in his daily work. Keyboard shortcuts provide one such performance enhancement. Currently I witness many migrations to the Open Source from Proprietary software products. They were accustomed to many of the shortcut methods they employed to get their task done. After their migration they get a performance hit because of not having a similar shortcut scheme. Let me give an example. In legal writing we use many capitals in between a sentence so very frequently, I need to capitalize to title-case many words originally written as normal lower case. As I encounter such words I would press Alt+F3 to toggle between lower-case Title Case and ALL CAPS in MSOffice. KOffice would show up the find dialog and OpenOffice would do nothing on that key-press. So here my productivity is hampered.

There is a comfortable solution for this particular issue:

- a.) Let's create multiple scheme of shortcuts for users of competing Office Suites.
- b.) In the configuration dialog we can have a combo to select from the active shortcut scheme.
- c.) For features like the cited example where we don't have a parallel feature still available, we can have a script implemented linked to the shortcut.

[Back to Contents](#) 



Wizards and Druids

Usability	(★ ★ ★ ★)
Audience	All New and Enterprise Office Suite Users
Necessity	High
Implementability	(★ ★) - Relatively easy

Currently there are few wizards and druids available for the KOffice users. This primarily is an disincentive. New users as well as Enterprise users need wizards. Specifically designed wizards speed up a user's output. Wizards should provide for graphical clues like a sample preview pane for the user. Wizards should show explanatory text about the options.

Wizards should commonly be available for the following functions:

- a.) A mail wizard for creation of Professional Letters and auto text based mails(most of the content is pre-written as fragments of text and these portions are dropped on the letter during composition (Generally used by Enterprise/Corporate users).
- b.) A mail merge wizard for Mail Merge of Letters. (Currently mail-merging to e-mail is not supported. I feel this is a very necessary requirement in the face of digital communication revolution.)
- c.) A Presentation wizard for creation of Presentations through selecting options.
- d.) A Slide show wizard with preflight check for Presentations.
- e.) A Chart wizard for creation of charts in Spreadsheets.
- f.) Scenario wizard for Spreadsheets.
- g.) Pivot Table wizard for Spreadsheets.
- h.) Goal-Seek wizard for Spreadsheets.
- i.) Wizard to Back-up (Export) and Import Program Settings.

[Back to Contents](#) 



Backing up of Program Settings

Usability	(★ ★ ★ ★ ★)
Audience	All Office Suite Users
Necessity	Very High (Not having this feature causes wastage of efforts)
Implementability	(★ ★) - Relatively easy

Note: To be read along-with Menus Toolbars and Shortcut Menu [↗](#)

When a user interacts with the Application for relatively a long time, he develops several shortcut methods for himself, creates templates for his regular use and changes the default settings to suit his needs. All these settings get lost when an upgrade or re-install occurs. All those settings and templates may have to be re-done or backed up by manual means but this is not a potential solution. Something more need be done.

Some Ideas for Backing Up Program Settings:

- a.) The Settings menu should provide an option to Export(Back-Up) and Import Program Settings.
- b.) This option should launch a wizard.
- c.) The wizard should list an assortment of items which can be backed-up for later use.
- d.) A comprehensive list is here:
 1. Created Templates
 2. Program Colours
 3. Default Document Settings
 4. Preferred Fonts (any other fonts from the pre-installed ones if used in the Custom Templates)
 5. Custom made Ordered list templates
 6. Custom Toolbars made
 7. Default Toolbar schema
 8. Any Extra Files . . .
- e.) The Wizard would then ask to select all or custom select components to back-up
- f.) After the selection is done, the wizard would ask for a file Name/Location.
- g.) After the Name/Location is provided, Clicking on Export Settings would create a file (generally a compressed archive) as output, at the destination.



Some suggestions for Importing the previously backed up Settings:

- a.) As stated earlier “Settings” menu should provide an option to Import Settings.
- b.) The Import option should open a wizard.
- c.) A File selection box along with some helpful text about Settings Import should be provided at the first page of the wizard.
- d.) After the file is selected; a button to verify the archive should appear.
- e.) The verification results should show whether the archive is in good condition and it should also show the contents of the archive in the following manner:
 - 1. This Settings archive Contains -
 - 2. Five Templates in Two categories
 - 3. No Colour Settings
 - 4. Custom Default Document Settings
 - 5. Four Fonts
 - 6. Two Custom Ordering Template
 - 7. One Custom Toolbar
 - 8. No Default Toolbar schema
 - 9. No Extra Files
 - 10. Would you like to import these Items?
- f.) Then Two buttons one with Custom select Components and the other with a Yes should be present to be selected.
- g.) Selecting Custom select Components would show a similar selection dialog as during the Export operation.
- h.) For preferences related to program settings like Colour, Default Toolbar Schema, the program should validate if these settings are importable for the current version of the program or not and appropriate message be shown.
- i.) After the job finishes, any success/failure message should be shown.

[Back to Contents](#) 



Start-Up Screen

Type	General GUI Topic
About	A Start-Up Screen for KOffice
Usability	(★) - Nil
Audience	All KOffice Users
Necessity	Fancy Feature
Implementability	(★)

A Start-Up Screen for KOffice

Currently KOffice is quite fast to load. Still I feel that as most Office Suites display a start-up screen, it would be a cool thing to do for KOffice. As long as KOffice remains fast to load possibly there is less use for it. It will vanish even fast from view. Start-up depends very much on what the application loads up initially. I would be much happier if KOffice loads up fast with a quickly vanishing KOffice Splash in a flash.

[Back to Contents](#) 



Menus, Toolbars and Short-cut Menu

Type	General
About	Enhancing Menus, Tool Bars and Short-cut Menus
Usability	(★ ★ ★)
Audience	All KOffice Users
Necessity	Usability Enhancement Feature (Making Features more Accessible)
Implementability	(★ ★)

Menus of KOffice Products

Currently KOffice has quite a manageable menu structure. However ways to enhance it should always be looked at. Clubbing of closely related menu items should be done and the Settings menu should be re-structured as it seems to be quite un-structured/inefficiently structured.

The “Settings” menu item should have the following structure:

- The Toolbars sub menu item should be moved under View. As the Toolbar item is always available through a right click on the Toolbars this is least used here and it is logical to find it in the View menu because it is about viewing/hiding the Toolbar.
- The “Settings” menus should have three sections.
- The top section is for frequently used settings or quick settings
- The mid section is for configuring KWord/KSpread etc. and for Global KOffice configurations.
- The bottom section is for importing and exporting settings.

A Sample Settings menu:





ToolBars of KOffice Products

Toolbars are visual short-cuts for menu items. They are designed to speed up feature access. As it is generally expected that people using KWord would be using other office products earlier, to make them feel at home the Toolbars should be schemed. That means: from the settings the user should be able to make the Toolbar look and feel as if it is his native program's tool bar. This also implies that from the settings, the Toolbars could be schemed to look as the Toolbar of OpenOffice and other office products as well as there should be an option to restore KOffice defaults. While modding the Toolbars, any user defined items should be left untouched.

Short-cut Menus of KOffice Products

Currently KOffice Short-cut menus has quite a manageable structure. However the visual clues are not quite distinct. In OpenOffice, it is overwhelming and the visual clues are not well-guided. Captioning in Short-cut in menus like we have in KPdf would be better to provide visual clues. However something better can still be done.

Here is a sample Short-cut Menus with captioning:



[Back to Contents](#)



Ruler, Guide, Tabs and Margins

Type	General
About	Object positioning in a Page Layout
Usability	(★ ★ ★ ★ ★)
Audience	All KOffice Users
Necessity	Usability Enhancement Feature (Fine Control)
Implementability	(★ ★)

The Ruler

The ruler is an important GUI element while viewing the page in "Print Layout" view. Apart from providing on-screen measurements, it should make us able to control text and graphic placement on the page. Currently the ruler of KOffice shows markings from the start of the page while OpenOffice shows markings from starting from the margin. Here I consider the approach of KOffice to be quite logical and well placed. I have done quite a good amount of page laying so I know that adding and subtracting from ruler values can be quite frustrating; as on a physical page you have the opportunity to take measurements only from the origin of the page and having a paper page and comparing the markings to the page in computer could be very frustrating if the ruler in computer does not follow the same way as the physical ruler. So Don't change this behaviour in future.

The ruler of KOffice does not show up the finer markings on the ruler so infusing a bit of dynamism would do plentifully good. On magnification, the ruler should show up the finer markings to help in finer controlling of the objects. On zooming out, the finer markings should become invisible to aid legibility. Further more, the ruler should have a double coloured representation so that it could be visually more distinct.

The ruler of KOffice does show up starting position and ending position of a text frame and other objects but these positions are not modifiable from the ruler. Having a small bar below the ruler to show up and to aid in dragging would be quite helpful. In OpenOffice the dragging of margins are made possible through a very narrow place between the hanging and paragraph indent sliders and the position is too narrow to exercise this option. When the paragraph indent is dragged the hanging indent is also dragged alongwith. There are no modifiers for this behaviour. By adding a modifier along with dragging paragraph indent (with Ctrl/Shift pressed) should not change the hanging indent. For visually modifying the object placement, I am

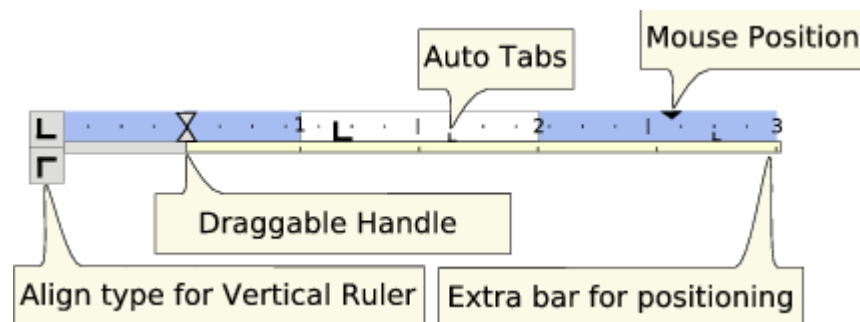


suggesting a bi-level and dynamic ruler.

The suggestion list for the ruler are:

- a.) Bi-coloured ruler for better visual representation.
- b.) An extra bar at the bottom of the ruler to aid in visual placement of objects.
- c.) Auto tab markings on the ruler should be visible like in OpenOffice.
- d.) The automatic tab settings should be changeable through settings.
- e.) Zooming in and out should dynamically show/hide finer markings on the ruler.
- f.) The mouse position should be shown on the ruler as a running marker.
- g.) A little beautification of the ruler would be welcome.

A sample implementation is here:



Guide

Currently Word-processors don't feature a "Guide" for object placement. However for DTP, it is a very essential tool. The possibility of using guides would be quite beneficial for publishing people. For a document which flows from top to bottom, horizontal guides would be less useful. However as KWord is a Frame based Word-processor, this limitation applies only to a particular set of documents where a main text area is automatically supplied which provides for such text flow and in other documents where the text boxes are manually added and linked a guide would be as useful as in a specialist Page-Layout application. As KOffice already possesses that speciality from long, it is time to exploit it for users needing special DTP tasks.

Tabs

Currently the Automatic tabs are not configurable for size in inch/cm or number of spaces. The Auto Tab positions are not visible in the Ruler. In the Ruler Section there is a



suggestion to make the Auto Tab positions visible. Further there should be a configuration setting related to the document for Automatic Tab positions.

Margins

In a frame-based word-processor like KWord the margins are the boundaries of the bounding textbox. Visual resizing is currently not possible in some kind of documents (Blank Document, Professional Letter etc. where the main text area is automatically added). In the Ruler section there is a suggestion to make visual resizing easy.

[Back to Contents](#) 



Client Area

Type	GUI Specific
About	The Client Area of KOffice Applications
Usability	(★ ★ ★ ★ ★) Productivity Enhancer
Audience	All KOffice Users
Necessity	Relatively High
Implementability	(★ ★ ★) - Moderately Hard (Multiple Features)

Non-Obtrusive Dynamic Worksheet

The usability of a worksheet is very much dependent on the services it offers. Here we are talking about a plain area, not a very graphically rich place. Classically the worksheet is supposed to be a plain and blank place to work on. However as the work loads as well as demands of new generation of users have gone-up manifold, there is a high need to infuse dynamism to this plain and simple world.

Several experiments have been made by software vendors in this field. Some ideas presented here are quite well known. Here is a specifically selected set of ideas for KWord.

1. Assisted Typing Technology (Mainly for Word Processor)
2. Help Bubbles (For all Office Products)
3. Mini Tool Bubbles (For all Office Products)
4. Worksheet Rigidity (Mostly For Word Processor and Presentation)
5. Automatic Obstruction Removal (For all Office Products))

Assisted Typing Technology

The main task done using a word processor like OpenOffice.org Writer or KWord is document preparation. For a document to be prepared the first and foremost task is to type it out. Many alternative means of inserting text has been tried in past and some are quite innovative too like Speech Recognition, Hand-Writing Recognition etc. but nothing beats the old typewriter. The keyboard is still the much preferred way to enter text in the documents. Small tools like recurring text blocks (AutoText), Simultaneous Spell-checker, Simple Typo Replacement scheme (AutoCorrect) and Special Function keyboard shortcuts go a long way in



helping the typist finish the work faster and with lesser defect percentage. AutoText, AutoCorrect, Simultaneous Spell-checker are much famous by now. However another tool needs special mention here. The concept of providing a suggestion of words from the current text is one such tool.

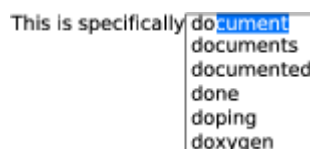
The concept of Suggestive Typing is not new. It has been used in OpenOffice.org and KWord but probably the better implementation is yet to come. The main criteria is when to begin suggesting and the minimum suggestion length. I have found out that most of the double and single letter suggestions are functionally quite similar to absence of any suggestion because Enter key is not the part of the standard typing keys and is a special key. Lifting any of the fingers to hit Enter is about twice as slow than hitting a normal key under the typing keys (Q to P | A to ; | Z to .). A point to be noted here that the symbol keys and extreme keys like Z are comparably slow than the keys directly under the index-mid fingers. Digits on the top row of the key board are even slower to hit than the Enter key. On a full sized key board the cursor keys are easier to hit than the Digit keys but on a Laptop keyboard they are harder to hit. This way the ergonomics of key layout have a say in the issue of Suggestive Typing. Currently, I found hardly any way to control the suggesting capabilities of OpenOffice. The “On The Line” suggestions have a distinct advantage for the eye. They are less distracting than bubble suggestions but they are handicapped in the same way because in this manner only a single suggestion can be given. Personally I find myself typing the whole thing most of the time because of a difference in the ending of the word. Unless it is quite a moderately long word (Understanding for example) I would never accept the text and hit BackSpace and correct the ending (I would accept Understanding and delete the trailing 'ing' to get Understand). There is a better implementation in the text editor of KDE. Many a times, I do the typing in Kate. The “KTextEditor Word Completion Plugin” does a better job by displaying a pop-up with multiple choices. Many a times, hitting the bottom arrow couple of times to select the word is less expensive than typing the whole thing out. Here the difference between the Eclipse/NB/VS style needs to be shown prominently. In a programmer's IDE a programmer selects from the drop-down whereas in a word-processor the typist knows what to type next. The problems get worse if you are creating a write-up on the word-processor itself. The IDE style suggestion bubbles can distract you much and can ruin your write-up by breaking your chain of thought. So the IDEAL solution should have the following properties:-

- a.) It should not be a tool-tip style bubble (No Back-Colour changes).
- b.) Soft edged/coloured bounding box with/without separators between suggestions; suggestions displayed using the same style as being used currently.
- c.) It should be On-The-Line as you type style (Least distraction and easiest to select).



- d.) It should be a list (Small, 5 is the productive maximum) with the first option set right on-the-line. Recently corrected miss-spelled entries should not find place.
- e.) It should be configurable for (Enable/Disable) and
- f.) for begin suggesting at 'n' characters and minimum length of suggestions.
- g.) Even if the typing goes on closing in the the suggestion-length, the suggestion should not vanish from the view till either the completion of word or invalidation of the suggestion. (Over rides the minimum length of suggestion)
- h.) Automatically disable suggestions for all levels of heading styles. (The real productivity of suggestive typing is on running text only).
- i.) After selecting a word from list by arrow both Enter and Spacebar should enter the word followed by a space. By selection by mouse the same should happen.

A sample of such an implementation:



Help Bubbles

Help Bubbles are very much context based. The appearance of the help bubble may be for displaying some help or for some specific text entry operation. These are designed to drag attention. These are the specific points related to the help bubbles:

- a.) The Bubble should appear during entry of special text like Month-Names, Day Names, Options to automatically enter blocks of text. They are supposed to appear non-obtrusively (to the top right for LTR and to the top left for RTL text entry) and vanish after their work is done.
- b.) The standard way to accept the bubble is to hit Enter and to reject the bubble is to keep on typing.
- c.) The bubble should be represented by a call-out with an arrow; the arrow pointing directly at the cursor. Rectangle bubbles generally don't tempt the user to accept the suggestion. The arrow makes it very distinct what the bubble tends to accomplish.
- d.) During scrolling of text a help bubble can be used to display current page number



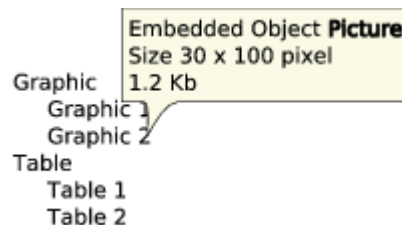
and chapter name with the callout pointing at the slider tab.

- e.) Such bubbles may ordinarily be used to show help text for context dockers during a hover over them or to provide tool tip like functionality.
- f.) These Bubbles should be capable to show very basic formatting like Bold/Underline and hyperlinking.

A sample implementation:



During Typing



During showing of object properties in document navigator.

Mini Tool Bubbles

Mini Tool Bubbles are small toolbars with or without some text. If at all there is any text that should support basic formatting like that of Help Bubbles. The main task of mini tool bars is to appear on context. They are designed to vanish after their job is done. Toolbar is a term quite large for them. They should contain no more than four active buttons. Buttons with drop downs should be placed to make them expandable to add more useful options. Some of the functions which can be added up as Mini Tool Bubbles are:

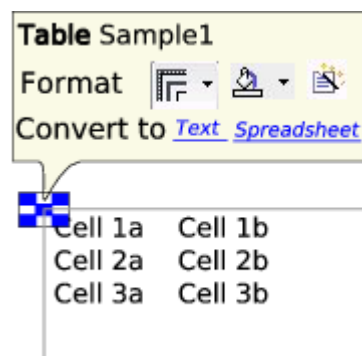
- a.) Paste Options like that of MSOffice
- b.) Field modification by hovering over any inserted field.
- c.) Header and Footer operation (Minimal) by hovering over the area.
- d.) Options for automatically corrected entries and to set exception for the automatically corrected items recently reverted back.



- e.) Options to change styles of ordered lists by displaying the styles already in use in the current document.
- f.) Options to edit objects like editing picture in a separate editor.
- g.) Options to modify links in the document or to follow the links provided in the document like an email address or a web-link in a separate browser.

By concept Mini Tool Bubbles are hover based. They are destined to vanish. They are not to remain on screen for long. The Mini Tool Bubbles should be controllable by an options page in the settings to make them enabled/disabled. This feature is specifically helpful for Word Processing and Spreadsheet tasks. Presentation software can also use them very effectively due to heavy use of multiple embedded content in them.

A sample implementation:



Worksheet Rigidity

The basic concept of dynamic worksheet is based on run-time modification of the interface of the application. However high fluidity in the client area of the office suite is one of the least desired elements as popping up bubbles, dynamically appearing tool bubbles and ever changing context based dockers are great distractions for creative writing and confusing at the first sight for the new users. So some rigidity is necessary.

- a.) Configurable dynamism (Options to disable/enable dynamic bubbles and auto changing dockers).
- b.) The typing area should not shift left/right or top/bottom on auto appearance of toolbars and change of dockers.



- c.) Soft appearance of context based Mini Tool Bubbles for embedded content/pasted content should not cause the embedded content to shift or vast chunks of text to hide.

Automatic Obstruction Removal

The dynamism of the dynamic worksheet may cause obstacles to appear in form of Mini Tool Bubbles, Help Bubbles and Toolbars. The basic idea of the bubbles or toolbars is to assist but if they remain on screen they may cause obstruction. So they should move out of way during typing.

Some suggestions on Automatic Obstruction Removal:

- a.) The Bubbles are supposed to appear non-obtrusively and vanish after their work is done.
- b.) The tool bars should automatically move out of the way of typing (differently for LTR and RTL text input)
- c.) And dockers should auto contract/roll up to clear the typing space.

[Back to Contents](#) 



List Handling

About	Handling Lists
Usability	(★ ★ ★) Usability Enhancement Feature
Audience	Word Processor and Presentation Application Users
Necessity	Moderate
Implementability	(★) - An Option Pane with Dock Integration

Lists that do not change are easiest to manage. However all documents are not pre-written. Many a times managers want the lists to be manipulated on a presentation slide. Some points need to be promoted and some needs be elevated to upper ranks in the list. Such manipulation is easy if the list is a bulleted one. If it is an ordered list then problems crop up which needs manual structuring. Believe it or not, it is one of the most boring tasks around. All of your efforts get concentrated on getting the numbers right than proper logical ordering of the list. It means that you don't concentrate on your creativity and instead focus on the mundane task of getting the formatting right.

OpenOffice provides convenient list management through its bullet and numbering toolbar. However the implementation is not perfect. Most of the time the promotion and demotion spoils the ordering. A proper implementation is always welcome.

Suggestions for List Handling:

- a.) Multiple type of Ordered List support. The current type of alphabetic list which I have extensively used, is not supported in OOo.
- b.) Options to custom configure an existing Ordered List.
- c.) Options to add (Region specific) Ordering support type. Generally a user would open an option in the list handling dock and start creating a sample list where it is defined about which characters follow which one and what changes happen when the list runs-out. Similarly for numerical ordering, local counterparts to the Roman numerals should be definable.
- d.) In the settings Import/Export (discussed else-where) There should be an option to back-up these custom ordered list templates.
- e.) Common marks like '.', '-', '>' and ')' should be automatically appendable to the orderings.
- f.) For Unordered lists like bulleted lists, the type of bullets should be choosable. Apart from the HTML bullets like Disc, Square etc., selecting a custom bullet from



a font (generally a symbol font like Dingbat) or from a graphic file should be easy. Recently used bullet types should be given priority while showing the custom bullet dialog the next time.

- g.) OOo have a good assortment of bullets and numbering but there are no ways to modify them. This restricts the user to eight varieties of bullet and numbering schemes. This is too meagre for professional documents or presentations.

[Back to Contents](#) 



Tables and Other Objects

About	Other Embeddable Objects in Office Documents
Usability	(★ ★ ★) Usability Enhancement Feature
Audience	Word Processor, Spreadsheet and Presentation Application Users
Necessity	Relatively High
Implementability	(★ ★) - Multiple Features

Apart from Lists there are several other embeddable content generally found in office documents. They are, Tables, Charts, Floating Text Frames, Scalable Graphics, Bitmap Graphics, Formula etc.

Several suggestions on handling of Embeddable content including Tables:

- An embedded object should be allowed to wrap text around using different text wrapping schemes.
- The object should be movable around using arrow keys and fine control should be obtainable on the placement using snapping to guides, grid or without any restrictions using modifier keys like 'Alt' or 'Ctrl'.
- Every embeddable object should be allowed to be bound by a surrounding box and the visible boundaries and the style and the width of the boundaries should be configurable through a Borders and Shading contextual dock.
- Anchoring is a good idea of placing objects in the midst of flowing text. Anchoring to paragraph or characters should allow for relative placement from that specific paragraph or character. However option to turn off anchoring (remove anchors) should be present. (DTP people prefer to arrange objects themselves). Anchors in OOO jump around in a berserk manner if the object is moved. This should not happen. Anchors should only move if the anchor marks are moved. Moving Objects when the anchor is set at some place should relatively position the object only.

Tables

Several Extra suggestions on handling of Tables:

- Automatic positioning of tables have never been satisfactory. This is the same with OOO and also the commercial applications.
- A sane default while creating a table is good but ability to change it as per the



user's need is extremely important.

- c.) The Length and width of table should be graphically definable. For very long tables only width can be graphically set.
- d.) Once the width is set by the user, any other adjustments in column widths should not make the table width to change. Often the table is seen changing size when the columns are resized in several applications.
- e.) Once a table is selected, the table context dock should be available. Options to define the cell alignments, column widths, row heights should be immediately available through use of icon combos and spinners.
- f.) Advanced table formatting like applying a table style should be provided as a link to the style gallery. Any selected style should be further customisable and the customisations should be preservable using “creating custom styles from selected table”.
- g.) Whilst changing of the text and background colours should be normally possible; options to fill the background with patterns and bitmaps should also be available.

Charts

Several Extra suggestions on handling of Charts:

- a.) Creation of charts should be wizard based.
- b.) Buttons for graphically selecting cells should be present.
- c.) Charts already made should be modifiable through a chart context dock.
- d.) Chart types should normally be changeable using a chart gallery.
- e.) Complete control over colours and graphics should be there.
- f.) Along-with the colours, filling of bitmaps and patterns should also be allowed for chart elements.
- g.) After a chart is embedded, graphically modifying the line thickness, line pattern etc. controls should be possible.

Pictures

Several Extra suggestions on handling of Pictures(scalable and bitmap)

- a.) Several formatting options for bitmap pictures should be present.
 - 1. Conversion to Gray-scale, Watermark and other common effects.
 - 2. Options to change contrast and colour.
 - 3. Options to change gamma value.



4. Slider to change opacity on the whole and eyedropper to make a specific colour appear transparent.
- b.) A button to reset all transformations should be present.
 - c.) For scalable images support for the following should be present.
 1. Proper support for SVG should be there. Currently Karbon's support of SVG is quite dismal. Support for Inkscape's custom extensions to SVG would be greatly appreciated.
 2. Options to add an outline based configurable shadow to the vector object would be appreciated.
 - d.) To make insertion of graphics a breeze, a Clip Browser is proposed.
 1. The Clip Browser should have links with the OpenClipart web site.
 2. The Clip Browser should support extracting vector graphics from Dingbat fonts. (Kindly view the suggestion on the discussion on Presentation [↗](#) why lacking support for extraction of vectors from Dingbat would be a great miss.)

[Back to Contents ↗](#)



Extra Context based Tools: Docks

About	Contextual Dock Bar
Usability	(★ ★ ★ ★ ★) Usability Enhancement Feature
Audience	All Office Suite Users
Necessity	High
Implementability	(★ ★ ★) - Multiple Features

KWord sports a bar to show “Document Structure” to the left of the document pane. Similar pane do exist in KPresenter to show slides. The sidebar of FireFox or Konqueror are also similar examples.

Several limitations of the current implementation:

- This area is quite static in nature.
- There is functional difference between a browser approach and the approach of a Word-processor or a Presentation Software.
- The side bar (dock) is fixed on left side whereas scientifically it should be on right side. (point b above)
- No extra option is available from the side bar.

This area can better be used as a contextual dock to the right side of the document pane. Options to drag it to the left should exist for people already comfortable with the current side bar. The whole concept of this contextual dock is to provide better accessibility to the program features. The dock should automatically change to provide current context based options. For example, when a picture is selected, options to manipulate the picture should appear in the context dock. In addition, if more than one dock becomes valid for the current selection, e.g. For a currently selected table, both table manipulation dock and borders and shading dock become valid simultaneously, in such a case both the docks should appear sharing top and bottom space in the area. Then both docks can be used together or each one may be allowed to fill the dock bar depending on the use. A drop-down at the top of the dock bar can be used to reveal all the contextual docks available and one can be explicitly chosen.

Several Extra suggestions for the contextual dock.

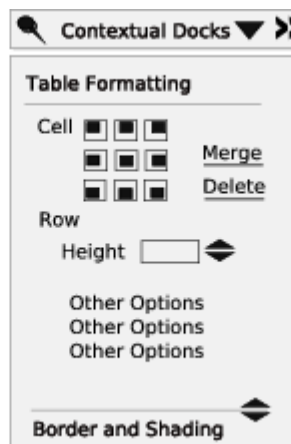
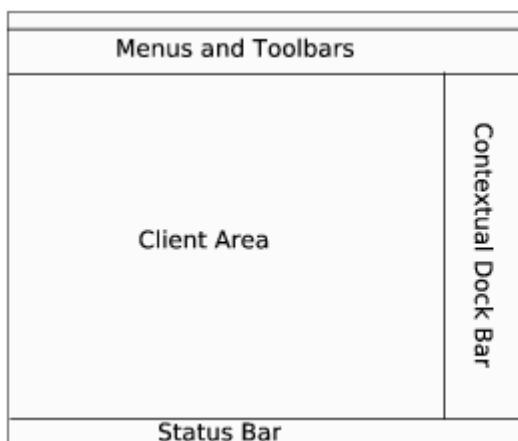
- The dock bar should be collapsible on a button click or pinnable for permanent



display.

- b.) The dock bar should be accompanied by a horizontal and a vertical slider. Thus even if the contextual dock does not fit in the space it can be made fully visible.
- c.) The dock bar should have a drop-down to explicitly select an available dock.
- d.) More than one contextual dock should be able to show-up in the dock bar.
- e.) Several uses of this contextual dock bar is scattered over this write-up.

A sample contextual dock:



Sample Contextual Dock for a selected Table. The Border and Shading Dock is at the bottom accompanied with a drag handle.

[Back to Contents](#)



Special Discussion on The Word Processor

Type	Application Specific
About	Various Topics relating to the Word Processor

Application Characteristics

A Word Processor is the most used application in an Office Suite. A Word Processor is used by all kinds of users because of its functions. From writing of a small letter to the creation of a big book, and every document in between needs a Word Processor to be productive. As this part of the Office Suite is the most important one, I have given special coverage on various issues related to the Word Processor in this write-up.

Improving KWord as a documentation Tool

Usability	(★ ★) High usability for Documenters only. Not for general users.
Audience	Documentation creators
Necessity	Generic, Useful for Fast and Quality documentation generation.
Implementability	(★ ★ ★) - Moderately Hard (Multiple Features)

A Word processor is specially beneficial while creating content. Currently Open Source Applications do need a lot of documentation. Tools like Kile can be good at creating the perfect documentation but won't be as productive as a Wordprocessor. Support for common documentation formats like docbook would be a welcome step in this direction where new documentation can be easily created and managed.

Printable user manuals are preferably created as PDFs and as QT4 series have good capabilities for creating PDFs, the same source can be utilised in creating printable content without much botheration.

Currently pasting HTML content from a browser loses all the formatting in KWord. This also happens if we try to open an existing HTML document. This situation should improve. HTML support is crucial for any documentation tool.

[Back to Contents](#) 



Content Creation and Basic Formatting

Usability	(★ ★ ★) Generic Feature
Audience	Writers and basic Word-processing Users
Necessity	Moderate
Implementability	(★) a minimalist shell for KWord.

A word processor shows up a graphically rich environment. This environment richness is not well accepted by the creative people, writers and basic users who don't need advanced formatting. Many content creators resort to a text editor for their writings. I have seen people do basic text formatting like bold and italics using codes in a text editor. The reason? The absence of a lean word processor with very basic formatting abilities only. Even from the current breed of word-processor users, many would prefer a lean version for content creation. There are two main attractions of a word-processor for a content creator, automatic spelling suggestions and common grammatical corrections. Extra language tools like a thesaurus are welcome additions.

Most page-layout applications employ a story editor to provide a clean interface for content creation purpose.

For small documents like a single page letter, most of the advanced formatting tools are not generally necessary. However for word processors switching from a minimalist simple view to a standard configurable view and vice-versa have never been implemented. This requires attention.

Font and paragraph styles are necessary elements to quickly perform a consistent formatting job. Many word processors club these separate things together to form styles. As a paragraph formatting have more scopes to change than a font style, this creates problems for the user. So either nested style creation is necessary or decoupling of the font styles from paragraph styles need be done.

Over the time the distinction between the word processor and a dedicated page-layout application have become very much blurred. Less advancement in the area of word-processing has happened. The Open Source word processors have supported spelling suggestions since quite some time but grammatical suggestions seem no where in sight. The best workable solution for grammatical suggestions is available is Microsoft's Word and that is too limited to call it a grammatical tool. This is a very sad situation.

Many people from the developer community have seen or used Eclipse Java IDE. It employs a concept called "Perspectives". Our own file manager cum browser Konqueror



provides view-profiles. A similar solution for the word-processor of the KOffice suite is needed with a perspective of a very minimalistic interface with quite basic formatting and full language controls. Further, no docks and no popping toolbars. This would be a boon for content authors and people who need very basic formatting needs for their documents. This interface should be changeable any time to the full-featured interface using an option in the Settings menu.

[Back to Contents](#) 



Page, Paragraph and Text Formatting

Usability	(★ ★ ★ ★ ★) Generic Feature
Audience	General Users
Necessity	High
Implementability	(★ ★) Restructuring work

Formatting page paragraph and text is the commonly available functions in a word processor application. While editing job is under progress in a word processor, proper support for such formatting is highly essential. Generally for text formatting the formatting toolbar is sufficient. However for formatting of paragraph and page, dedicated dialog boxes need be filled up.

An assortment of tips to enhance accessibility of these features with the use of intelligent options and contextual support:

- a.) A right click on the page outside the text box should show options related to page formatting.
 1. Link to Page Formatting dialog
 2. Options to enable/disable Header and Footer
 3. Options to enable/disable Foot Notes
- b.) While a text is being typed, selected or the cursor is placed in a paragraph (spanning multiple lines), a couple of paragraph related features should be available in the shortcut menus available on right click:
 1. Paragraph style
 2. Line spacing
 3. Alignment
 4. Bullets and Numbering
- c.) While the cursor is present on a line, the following options should be available from the shortcut menus:
 1. Changing of case (The current option of opening a dialog should be replaced with the options available as an expansion of the menu.)
 2. Changing of font size (A short list of sizes; both higher and lower.)
- d.) While on paragraph or a line the contextual dock should show a single dock with entries for both paragraph and text styles. (These different items should never be clubbed together.)



Multi-Level documents

Usability	(★ ★ ★) Generic Features
Audience	Writers and Book Makers.
Necessity	Moderate
Implementability	(★ ★)

Documents which are quite organised using headings, chapters and sub-chapters can be put in this category of documents. This creates a structure analogous to a multi-level list. Currently documents in word processors by default open in a page-layout view. This view is very productive only for shorter documents. For longer documents like this multi-level document needs a different type of handling.

Most programmer IDEs have implemented code folding. This enhances readability of the code by focusing on only the code under discussion and collapsing other code. Similarly as a multi-level document is a hierarchical representation, ability to do similar text folding is necessary. Microsoft's Word provides for an out-line view with similar functionalities but its accessibility features are limited to folding and unfolding only.

A document structure showing the document tree should be made available during the process of editing a multilevel document in the contextual dock. A tool bar designed to promote, demote and move up/down should be provided along-with for manipulation of the multi-level text. (Similar to The Bullet and Numbering Toolbar in OpenOffice)

A default font style scheme should be provided.

Possibilities to customise the font style scheme should be there.

Linking paragraph styles with font styles creates problems for the user. Allow keeping them separate.

Automated options to create table of content with/without links from a multilevel document should be implemented along-with.

[Back to Contents](#)



Multi Layered Documents (Master and Meta Documents)

Usability	(★ ★ ★) DTP related Features
Audience	Publishing Industry and Collaborative content creators.
Necessity	Relatively High (Parallel solutions are few and mostly commercial)
Implementability	(★ ★ ★ ★) Complex and multiple feature requirements

Master Documents

As the Word-processor Applications get user-friendly and powerful day by day the job of specialist page layout applications are getting reduced. KWord is a Frame-based word-processor which makes it a potential challenger in the Open Source DTP arena. It needs some power-booster however. Here we discuss about Master and Meta Documents which will help KWord to create professionally looking books, catalogues, brochures etc.

The concept of master document probably has been completely mistaken by the OOo team. What they call a master document is in reality a meta document. These meta documents are necessary for voluminous and/or collaborative text. Master document is a term from the DTP area. A master document is a template which can be applied to any document page. This is especially helpful in creation of books, catalogues and brochures. A master document generally holds repeatable text and graphics which are applied to specific and several pages. Generally they are used for custom made chapter beginning pages, with specially formatted and positioned Header and Footer text from the original document. Multiple masters layered one over another makes it easy to maintain repeatable graphics on pages. As KWord is a frame based word-processor, it can be added with powerful DTP oriented tools with relative ease. The ability to handle multiple masters would be an excellent addition for Open Source DTP solutions available for general mass.

Suggestions for Master Documents in KWord:

- a.) Multiple masters should be created and managed for a single document.
- b.) Application of Multiple masters over a single page should be possible in a layered fashion.
- c.) Master pages should support showing Chapter names, Chapter numbers, Author Name etc. meta-data from the document. Additionally these master pages should be capable to show meta-data in a meta document. As a meta document can



contain such meta from its constituent documents, there should be means to show/hide the meta data from the constituent documents.

- d.) Cloning of Masters should be possible.
- e.) Assigning names and numbering of masters should be possible.

Meta Documents

Meta document is a collection of documents. A meta document provides a single shell for them. It also provides an independent layer for adding annotations, remarks. It is the perfect means to create abstracts from existing documents complete with annotations, suggestions for corrections, explanations along-with portions from the documents referenced from it; all these with the additional safety of not affecting the integrity of the original documents. A meta document should make it possible to apply different styles to the referenced text so as to minimise interference with the original documents.

Target Audience:

Publishing industry is the best guess. However there are other users too. They are people like Section Officers, Head Clerks, Report Evaluators and Mid-level management staff. The format of such a meta document is generally a text document but it can-not be limited to it. Another meta document which we generally use knowingly/unknowingly is the presentation software.

Note: A Spreadsheet would be too complex to be a meta document. Even if created, it would be difficult for users to effectively utilise its functionality. The reason: a spreadsheet does not lend itself well to multi-object in a single page. Even the charts we create don't embed by displacing content from bottom. They just float over the sheet. A meta spreadsheet could only be a container for multiple spreadsheets.

A meta spreadsheet can become an excellent analysis tool. Further more the meta spreadsheet could be made a part of a meta text-document.

In any meta document the documents referenced are independent and so changes can independently take place in them. So there should be possibility to create snapshots (described later) of the current status of the document for comparing it with a later version. The meta documents should be made exportable as a package (snapshot creation). During creation of a meta document multiple sources are referred. As referenced sources are dynamic pieces of



information and are not expected to be available locally always, when we try to view it at a different location the referenced pieces may not show up (another reason for creation of snapshots). So there should be means to embed the referenced data for viewing it at a different place. This is the creation of a snapshot. Tools to automatically compare a snapshot with another snapshot or with a current version of the document is expected along-with. This would work like a diff for meta-documents.

Usability of this feature

The average home/casual user is not expected to utilise this functionality. The power user in the Government offices, companies and in the publishing industry are expected to make the most of this feature. For creative writers and authors of technical manuals it would be quite a helpful tool.

A meta document will be a collaborative tool for the publishing industry and a book authoring tool for writers. Combined with the layout of the master document, the whole process starting from writing a text, verification and editing, page-layout and pre-press task can be performed on the same text that with the advantage of working with the same application. This is the practical usability of this feature set.

[Back to Contents](#) 



Professional Pagination and Pre-Press work

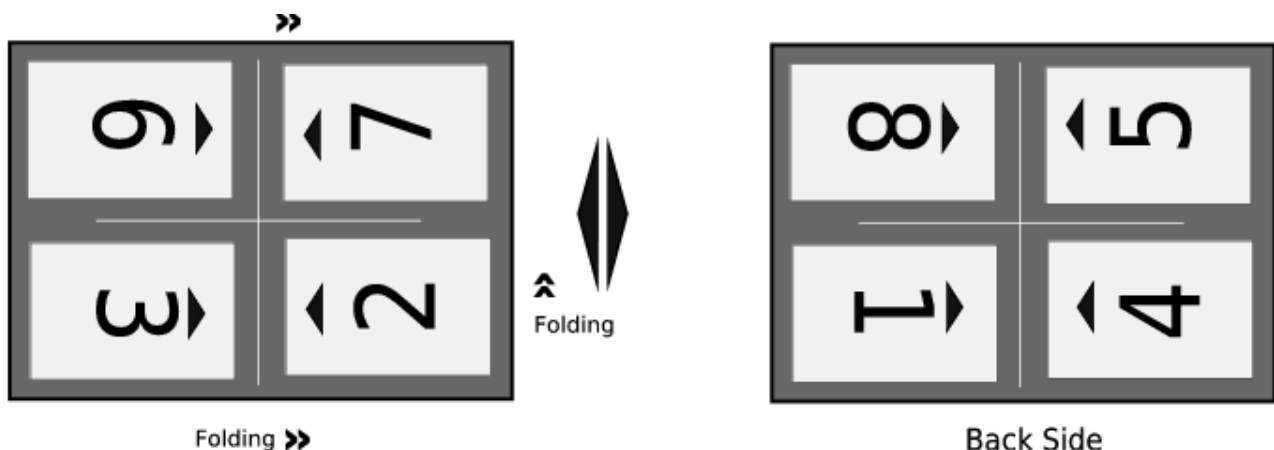
Usability	(★ ★ ★) DTP related Features
Audience	Publishing Industry
Necessity	Relatively High (Parallel solutions are few and mostly commercial)
Implementability	(★ ★ ★) Complex feature requirements

Generally the pages we get from a computer print is quite different from the needs of the publishing industry. There are many more requisites. Pagination, which is limited to page numbering in the mind of any average user is a quite different story for the publishing industry.

Books are generally bound through stitching up of several packs of multi-side papers. The papers on which the printing is done is often quite large in size. The larger page is then folded and stitched together in bundles of specific quantities. These bundles are then stitched together bundle after bundle to produce a bound. This is quite different than printing on single side or double side and it has nothing to do with page numbers.

Professional Pagination requires a finally formatted document as input. All the pages starting from first to last is custom arranged on a pre defined sized sheet. Generally for small and medium size books (standard book size) a single sheet of paper contains four pages on each side of the paper. This totals to eight pages per paper. However each side containing four pages is printed on a single master for replication.

Here is a sample pagination:



This sample is only valid for a bundle containing a single sheet and generally that is not the case. Commonly two to four sheets are used as a bundle. In this case preparing the masters manually is quite a daunting task. This process can be automatically done through a Pre-Press



wizard. The wizard will ask about the target sheet sizes its cutting areas and gutter. Then it would expose a page sorter with a view similar to the above image and options to keep particular pages blank or importing pages from a different document should be allowed.

The next step in this wizard should show up a preview of the book with binding marks, folding marks and other calibrating marks. The user should be able to preview the output by scrolling page after page (also sheet bundle by sheet bundle in a side pane).

After preview of the book a preflight statistics should be shown about the following items:

1. No. of pages
2. No. of sheets
3. Fonts referenced
4. Number of graphic elements
5. Problems if any (Off page objects etc. stuff)

Finally the output file selection dialog should open and there should be option to save a Pdf in High/Low quality (mostly for graphics). Options to directly print to a printer should also be present. As with most independent applications, options to save the pagination should be provided to save for later review and/or use.

[Back to Contents](#) 



Special Discussion on Spread-sheets

Type	Application Specific
About	Rigidity needs for the Spreadsheet Application
Usability	(★ ★ ★ ★ ★) Practical Usability in a work scenario
Audience	Enterprise Users
Necessity	Relatively High
Implementability	(★ ★ ★ ★) - Moderately Hard (Adaptive and Multiple Features)

Application Characteristics

A Spreadsheet Application is not like just another application. There are fewer and more experienced users who use them. The users of this tool generally don't experiment around. Even after the raging success of OOO in Office Suites, this situation has not changed significantly. Mostly the serious customers of this application are from the enterprise and they are quite experienced with commercial offerings. As we all know a switch-over decision for the enterprise comes at great efforts because outlandish changes are not quite well accepted if the returns are not in the same measure.

Focusing on the Target Audience

The Spreadsheet application is considered as a high end application which require some advanced exposure to the computing. Mostly Small Offices, Institutions upto the higher Enterprise users are the target audience of this application. The jobs done by these people are report generation, charting, financial forecasting, financial analysis, creation and maintenance of official schedules. This certainly is not a home-user application and so the average home/casual user is not supposed to use this product regularly.

The Present Market

Microsoft's Excel is very deeply entrenched in this portion of the market. The use of competing commercial applications like Quattaro and 123 is far and few. Approximately six years earlier it was not the situation. There used to be a lot of 123 users around and now I see that most have migrated to Excel. This market is very rigid and is not quite tendentious to accept changes



as easily.

Scope for changes

Very little scope exists in this area for intuitive thinking and outlandish application building. The new user will raise multiple questions like whether it is compatible with application xyz which currently I am using? Is the syntax of formulae and references are the same? (he is asking whether they are exactly the same!) If all the functions and formulae in application xyz is present here too? What about function assistance, scenario, goal-seek? Yes, the scope is that rigid. Most existing users will demand the exact feature set and as we are talking about the users from Offices and the Enterprise, they are paying customers and they are quite demanding. To make an Open Source application succeed in this scenario is quite a big task. However it is easy to attract completely new users/learners towards this Open Source product; but this does not solve our purpose. If we are thinking of a Spreadsheet application getting accepted, we are certainly thinking about migrations. As I have already pointed it out, very less scope for intuition remains here. What we need is a feature by feature replacement. Any extra features can wait.

A set of suggestions for the “Spreadsheet Application”:

- a.) Good Import and Export of data. This is very crucial.
- b.) The number of functions should equal or exceed the commercial offerings.
- c.) Function assistance should be implemented.
- d.) Automatic Cell Identification by multi-coloured boxes.
- e.) Distinct visual clues while moving, copying, and filling cells.
- f.) Intelligent cell selection by mouse while editing a formula. (The cell references should be changed only if cursor is placed between a range or over a cell reference. Avoid accidents and use wizards and cell selection buttons.)
- g.) Cell fill and paste should give options through Mini Tool bubbles (discussed elsewhere in the Client Area section).
- h.) References beyond current Sheet should be similar with other commercial and Open Source offerings. Better still to be able to use more than one style of sheet references.
- i.) The blue arrow in the cell is ambiguous. It feels as if it would show up some secret



once we hover our mouse over it. Just don't do away with it. Try to replace it with a blue vertical bar exactly at the same place. The mark is helpful.

- j.) Comments show up in a tool tip. It could be better represented by a help bubble (discussed elsewhere in the Client Area section)
- k.) The Insert External Data should open an import dialog. There are loads of items in the list but common-use sources like Mozilla Address Book, SQLite, Kexi, UnixODBC are missing.
- l.) Charting should allow for preview before committing. Extensive formatting options for border, shadow, colours, gradients, fonts etc. should be provided to make the charting tool useful in real work scenarios.

Implementing all these features would require extensive work. However for the special status of this particular application, such effort only can help KSpread to make a mark in the commercially dominated market.

[Back to Contents](#) 



Special Discussion on Presentations

Type	Application Specific
About	Making the Presentation Application more user-friendly
Usability	(★ ★ ★ ★) Usability features in practical considerations.
Audience	General Users
Necessity	Relatively High
Implementability	(★ ★ ★) - Moderate (Multiple Features)

Application Characteristics

A Presentation Application is an application which is generally widely used. Not only corporate/enterprise users use them but also general users like teachers and demonstrators in schools and colleges use them as a teaching aid. Tremendous amount of scope do exist here for intuitive thinking and application enrichment. However the current state of the KOffice presentation software need several necessary features before it can go for off-track advanced features and usability enhancement.

Core Features

At the outset what The Presentation software needs is Templates, Not just the few supplied ones but lots of them. Nested templates (A master graphic and/or a text combo along with different type of body areas) are very much necessary. Wizards should be there which can take care of creating a presentation from selecting a template to adding text, lists and adding transitions to finally provide us a sample presentation which we may modify here and there to get to the final desired shape.

A presentation software needs fine control over transitions. KPresenter could not make a letter by letter and word by word transition effect. It even was unable to show a prepared .odp from OpenOffice.

The interface of KPresenter is very clean and it is a big plus. The interface of OpenOffice leaves very less space for the main slide. The dock bars of OpenOffice take a lot of space and they won't hide/auto hide. Such dock bars for KPresenter are welcome with caution. They should be able to hide/auto hide with options for pinning them.



A set of suggestions for the “Presentation Software”

- a.) A cleanly laid out interface.
- b.) Slides in preview should not consume a lot of space.
- c.) Collapsible contextual dock bars (discussed elsewhere).
- d.) Document Navigator should provide snippets from text.
- e.) Click to type functionality is required which will auto insert text blocks.
- f.) Intelligent list handling. A OOo styled Bullets and Numbering toolbar would be a better addition.
- g.) A Clip browser for adding of clips is urgently required.
- h.) The Clip browser should have the ability to extract small pictures from Dingbat fonts (fonts with many pictures). Note: Extracting a picture and inserting a symbol are quite different tasks and ignoring dingbats would amount to missing a great lot of freely available scalable pictures and symbols available as dingbats.
- i.) The slide show dialog should show translucent controls for Back, Forward and Pause alongwith Pen. Selecting Pen should automatically pause the show. The pen colour should automatically be suggested from a fluorescent set contrasting the majority background colour. Option to completely disable these controls should be there in the presentation preflight (required for automated presentations).
- j.) The Configure Slideshow is too limited for now. A Slideshow wizard should be implemented alongwith. The Configure Slide show should have all necessary elements like transitions, Show/Hide On-screen Slideshow controls, Manual Timing, Rehearse Timing etc.
- k.) A preflight report should show the number of slides, estimated time of the show (if automated) and number of parts of show dependent on manual control and externally referenced material present in the show like (externally linked Clips, audio, original fonts etc.). It should give you an option to package the Slides with all text and symbols as curves and all externally referenced material embedded in the file for transport to a different location for screening.
- l.) Embedded or referenced media elements like Audio Clips (not background audio or voice over) and Video Clips should be controllable(Play-Pause-Stop-Restart) with both on-screen and off-screen controls during the slide show. These controls should not interfere with Slide-show controls.



KPresenter currently is in a very nascent stage. Many advancements are needed to bring it to the level of competing solutions from OpenOffice and commercial offerings. However a presentation tool is productive if accompanied by a simple and non-obtrusive interface layout even if it lacks several animation and other features. As the users of this segment vary largely and belong to very different categories, first level users (like educators) not interested in power-features would certainly appreciate a clean and neatly laid-out interface of KPresenter.

[Back to Contents](#) 



Document Security, Encryption and Signing

Type	Document Security
About	Encryption, Digital Certificates and GPG

Selectable Encryption Support

Usability	(★ ★)
Audience	All Advanced Office Suite Users
Necessity	High for Enterprise and Serious Users
Implementability	(★ ★) - Moderate (Multiple Features)

Till now we don't have the encryption selection capabilities even for OOo. KDE supports many kinds of encryption and so KOffice is also expected to utilise this functionality. The Save and Save As dialogs should have an option to password protect the document just like OOo. After selection, in the password dialog, it should have an advanced button which should lead us to the encryption selection by offering an assortment of encryptions schemes to select from. The selection should also have key strengths displayed alongwith if not changeable from a drop-down there.

Options to utilise GPG is dealt-with along-with Digital Signing.

Digital Signing Support

Usability	(★ ★)
Audience	All Advanced Office Suite Users
Necessity	High for Enterprise and Serious Users
Implementability	(★ ★ ★) - Moderate (Multiple Features)

Digital Signing is the realm of Advanced users. However many general users have interest to have it utilized. Digital Signing is an easy way to identify the originator of the document. OOo supports Digital Signing by Digital Certificates. Now this becomes a tool for a very niche clientèle. Possibly most general users won't bother about getting a digital certificate from an authority. However for compatibility reasons with OOo and for corporate customers, support for Digital Certificates is a necessity.



For long GPG has been the preferred way to sign e-mails and sign and encrypt files. If we can utilise signing with GPG for Office documents, it could really have become a tool for masses. Inbuilt support for GPG for encryption in the Save dialog (refer to the previous [✎ Selectable Encryption Support](#)) would be a boon for small user segment which is quite large. Currently GPG would convert the file and render it un-readable for even signing and that is something we don't need. There should be some way to append the signature digest to the file without corrupting the readability. After-all the OASIS formats are archives. Even if OOo does not support GPG, to make the file readable a small command-line utility may be created which could then be used by people through an OOo macro to verify the signature. A GPG encrypted document can be made to retain its extension so that KOffice opens it rather than KGpg.

Suggestions for Signing and Encryption:

- a.) Selectable encryption schemes for the document file.
- b.) Support for GPG encryption for document security.
- c.) Digital Certificate support for signing the document.
- d.) GPG support for signing for masses.
- e.) Collaboration with OpenOffice people on this issue.

[Back to Contents](#) 



Montage and Show Packaging

Type	General
About	KOffice Distribution
Usability	(★ ★ ★ ★) Availability and Integration
Audience	New KOffice Users, Vendors and Solution Providers, Enthusiasts.
Necessity	Moderate
Implementability	(★) - Very Easy

Reaching the Masses

I am very much aware that Team KOffice does not distribute binaries as a strategic decision. However here I would like to point out a very important thing; "Source Code don't run" (Oops! what a great revelation!). Most of us are aware of it. Has anybody given a serious thought over it? Possibly not. Seriously, it is not a joke. It is a potential problem. (While saying this I am very well aware that third parties support binaries of recent stable and beta releases. Recently I completed downloading of the KOffice-Beta 1.5 from one such mirror.)

KOffice is intended to be an end-user product. Most end-users will prefer their software to be a click and run affair. Most Linux distributions do the packaging work but KOffice is not like the other applications. It is a suit of productivity applications which complement each other. Many people I know who don't run KDE don't run KOffice too. Reason? It may seem quite outrageous but it is true. They believe that KOffice needs KDE and KDE is huge. These are serious allegations and we need a non-technical and down-to-earth answer for that.

KOffice should be a complete package in itself. Dependencies need be resolved at the source of distribution (What the hell!). Many people have allergies with statically linked packages but they are real-world solutions. If that is not possible then a package with all the dependency resolving packages should be available for non KDE users. Here I mean a single file to download. It may be an archive or a self extractor or an installer it does not matter (not a Klik CramFS. This would take time and space to explain why not Klik. The concept of Klik is good but it is not a regular solution. Kindly refer the FAQ). How about a downloadable CD .iso with an assortment of documentation and OpenClipart clips. Caps, T Shirts, Commemoration Bags etc. should I go further? The summary of the story is "A single binary of KOffice capable of running on multiple



distributions and the binaries should be made available with or without distro specific add-ons (just like the recent OOo release; all inclusive even with KDELibs). And there are other items, KOffice goods, software and softwear. They are very necessary elements to spread the word.

We have very much parallel and competing PIM solutions on both KDE and Gnome. Many KDE users use Ximian Evolution as their PIM. KOffice should be able to interface with the Gnome PIMs along with KDE PIMs.

Many would argue that such an action would result in feature bloat, inflation of download size but it would really make installing, distributing and using KOffice much easier than that of now; take my word for it. This is a positive step towards commercial hardening too. These characteristics would widely enhance the displayability and availability of KOffice ultimately. Yes! we know Team KOffice won't do it as a strategic decision. Possibly we can forge some alliance with some organisation/vendor to do the dirty work. Nevertheless it needs to be done in right earnest. Many distributions would still offer their customised packages but many more smaller distributions would welcome the step. The standard binary thus made available should run on most well-known distributions perfectly. For most other KDE based distributions there are very few problems (if any) ahead.

Here we have discussed three important issues:

- a.) Availability of a single standard binary for multiple distributions with dependencies for easy download/distribution and use.
- b.) Integration with PIMs from competing software camps.
- c.) Spread the word with items other than software.

[Back to Contents](#) 



Concluding Remarks

KOffice is not just an other Office Suite. It is the only “Open Source Office Productivity Alternative” with the widest variety of applications starting from Word Processor to Vector and Raster image editors. Apart from OpenOffice, this is the only other “Open Source Office Suite Alternative”. The strength of KOffice lies in its quick loading and simple operation. Recently OOo have tried to make their interface much simpler and easier to use. Thus similar attempts to logically reorganise KOffice applications need be undertaken. A competition like this would be helpful in gathering ideas in this direction.

In my analysis of individual applications, I have found that KOffice as a whole should be improved in different ways before concentrating on individual applications. That is why, there are many general topics in this current write-up.

The adoption of OASIS file formats is a step towards file format standardisation. However several issues need be addressed here too. Whether OASIS file formats would support the requirements of “Master and Meta Documents” ([↗](#)) remains to be seen. Further more, no Office Suite have committed to address the dreaded font issue ([↗](#)). These are serious issues and they need practical answers.

A DTP solution cannot succeed if it fails to fully interface with a standard Office Suite. As KWord already possesses the basics of a page layout application, it can be transformed in to a very decent DTP solution. Perspectives like those of Eclipse will keep the advanced tools hidden for a minimalistic view and a DTP specific perspective can be easily loaded up to use the application in its full vigour. As Linux DTP is lagging far behind its commercial counterparts, something need be done soon. (Note: I have tested Scribus and found out that it is not adequate.)

There are many common expectations from a [Presentation](#) software and I have put together several of them from the user's point of view. A similar analysis of the [Spreadsheet Application](#) have also been done and the suggestions from an end user's point of view have been put forth.

Giving special emphasis on document security, I have dedicated a complete chapter for deliberations on the issue.

Wherever necessary, I have specially pinpointed the needs of the target audience of the application.

It is earnestly hoped that this exercise would help the guided development of the specific components of KOffice.