



Office Productivity Suite Competitive Analysis

Martijn W.H. Dekkers

Document License and Copyright

Copyright © 2002 Martijn W. H. Dekkers.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with the Invariant Section being "Preamble", with the Front-Cover Texts being "Original © 2002 Martijn W.H. Dekkers", and with no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

History

January 2002

Initial version

Acknowledgements

I would like to thank Josh Berkus from Aglio Database Solutions for his valuable editing work and suggestions, Gianluca Turconi for his untiring work for OpenSource, and Louis Suarez-Potts for his assistance and overall Good Work™ in the OpenOffice community. This document is a first draft, and not yet complete (if it ever will be) suggestions, feedback, comments, additions, etc. are welcome, and can be directed to mdekkers@openoffice.org. A more detailed description of the Corel suite is in the works, and will be included in a future version.

The latest version of this document can be found at <http://www.ooodocs.org>

Introduction

This document discusses the different major Office Automation suites available today. It delivers a methodology for selection of one of those suites, using a generic set of architectural selection criteria that are applicable to almost any business environment. These selection criteria have been drafted from a customer/consumer perspective, as opposed to a software vendors perspective, and will at all times protect the customer's best business interests.

The selection of Office Automation Suite is a task that usually affects one of the core productivity areas of any given organisation, and many business and IT managers are hesitant at best to "rock the boat" in this area. Users have been trained on a given product, know their way around within this product, and a multitude of documents have been created in this product's (usually proprietary) file storage format. Hesitation in this area is therefore justified. However, any business leader must continuously ask questions like:

- Can I serve my business in a better way?
- Has technology progressed in a way that allows me to improve on the bottom line?
- Can I lower operational costs?
- Is my business exposed to any threats?
- Am I restricted in my options when selecting vendors for products I depend on?

If the answer to any of these questions is "Yes", the business leader is compelled to take action. The question is what the best course of action would be.

Vision and Goals

The Open Group has published a set of architectural goals that can be applied to almost any business¹. These goals can be used as guidelines against which any proposal or IT implementation can be compared. These goals can be considered to be the overarching design principles for almost any IT-related decision:

Improve User Productivity

User productivity improvements will be realized through the following objectives:

Consistent User Interface:

A consistent user interface will ensure that all user accessible functions and services will appear and behave in a similar, predictable fashion regardless of application or site. This will lead to better efficiency and fewer user errors, which in turn may result in lower recovery costs.

Integrated Applications:

Applications available to the user will behave in a logically consistent manner across user environments, which will lead to the same benefits as a consistent user interface.

Data Sharing:

Databases will be shared across the organization in the context of security and operational considerations, leading to increased ease of access to required data.

Improve Development Efficiency

The efficiency of development efforts will be improved through the following objectives:

Common Development:

Applications that are common to multiple business areas will be developed or acquired once and re-used rather than separately developed by each business area.

Common Open Systems Environment:

A standards-based common operating environment which accommodates the injection of new standards, technologies and applications on an organization-wide basis will be established. This standards-based environment will provide the basis for development of common applications and facilitate software reuse.

Use of Commodity Products:

As far as possible, hardware-independent, off-the-shelf items should be used to satisfy requirements in order to reduce dependence on custom developments and to reduce development and maintenance costs.

Software Reuse:

For those applications that must be custom developed, development of portable applications will reduce the amount of software developed and add to the inventory of software suitable for reuse by other systems.

Resource Sharing:

Data processing resources (hardware, software and data) will be shared by all users requiring the services of those resources. Resource sharing will be accomplished in the context of security and operational considerations.

¹ See <http://www.opengroup.org/togaf>

Improve Portability and Scalability

The portability and scalability of applications will be through the following objectives:

Portability:

Applications that adhere to Open Systems standards will be portable, leading to increased ease of movement across heterogeneous computing platforms. Portable applications can allow sites to upgrade their platforms as technological improvements occur, with minimal impact on operations.

Scalability:

Applications that conform to the model will be configurable, allowing operation on the full spectrum of platforms required.

Improve Interoperability

Interoperability improvements across applications and business areas can be realized through the following objectives:

Common Infrastructure:

The architecture will promote a communications and computing infrastructure based on open systems and systems transparency including, but not limited to, operating systems, database management, data interchange, network services, network management and user interfaces.

Standardization:

By implementing standards-based platforms, applications will be provided with and will be able to use a common set of services that improve the opportunities for interoperability.

Increase Vendor Independence

Vendor independence will be increased through the following objectives:

Interchangeable Components:

Only hardware and software that has standards-based interfaces will be selected, so that upgrades or the insertion of new products will result in minimal disruption to the user's environment.

Non-proprietary Specifications:

Capabilities will be defined in terms of non-proprietary specifications that support full and open competition and are available to any vendor for use in developing commercial products.

Reduce Life-Cycle Costs

Life-cycle costs can be reduced through most of the objectives discussed above. In addition, the following objectives directly address reduction of life-cycle costs:

Reduced Duplication:

Replacement of isolated systems and islands of automation with interconnected open systems will lead to reductions in overlapping functionality, data duplication and unneeded redundancy, because open systems can share data and other resources.

Reduced Software Maintenance Costs:

Reductions in the quantity and variety of software used in the organization will lead to reductions in the amount and cost of software maintenance. Use of standard off-the-shelf software will lead to further reductions in costs since vendors of such software distribute their product maintenance costs across a much larger user base.

Incremental replacement:

Common interfaces to shared infrastructure components allow for phased replacement or upgrade with minimal operational disturbance.

Reduced Training Costs:

Common systems and consistent human computer interfaces will lead to reduced training costs.

Improve Security

Security will be improved in the organization's information through the following objectives:

Consistent Security Interfaces for Applications.

Consistent security interfaces and procedures will lead to fewer errors when developing applications and increased application portability. Not all applications will need the same suite of security features, but any features used will be consistent across applications.

Consistent Security Interfaces for Users

A common user interface to security features will lead to reduced learning time when moving from system to system.

Security Independence:

Application deployment can use the security policy and mechanisms appropriate to the particular environment if there is good layering in the architecture.

Improve Manageability

Management improvement can be realized through the following objectives:

Consistent Management Interface:

Consistent management practices and procedures will facilitate management across all applications and their underlying support structures. A consistent interface can simplify the management burden, leading to increased user efficiency.

Reduced Operation, Administration and Maintenance Costs:

Operation, administration and maintenance costs may be reduced through the availability of improved management products and increased standardization of the objects being managed.

Consistent application of these goals will lead to a greatly enhanced IT environment. For more information, please visit <http://www.opengroup.org>.

Problem Statement

This section looks at two problems affecting almost every organization that uses IT – Supplier Independence and Software Licensing.

Supplier Independence

In terms of desktop dominance, the position of Microsoft in the market place is a foregone conclusion. Most organizations have standards in place that dictate the use of Microsoft Office and Microsoft Windows on their office desktop. While these policies seem to make things simple, they ignore a significant risk. At the moment, the successful operation of those desktops is dependent on a single software vendor. This section looks at the disadvantages of excessive dependency on a single software vendor.

The concept of supplier independence is often perceived as real competition, in the sense that we – as buyers – are in a position to select products or services from a number of different suppliers. Total dependence on a single supplier is usually perceived as a monopoly situation, and is naturally undesirable. Every rule and policy that governs the generic procurement process is founded on the principle that there are multiple suppliers for a given product or service, and that real competition exists between them.

Elementary economics teaches us that monopoly situations tend to lead to slower, less significant product development, poorer support and services and higher prices. Today, some vendors have a virtual monopoly on software development. It is a known fact that real competition spurs development, and leads to a range of products that are a better match to user requirements.

This document makes the assumption that plurality of suppliers is desirable, since this gives a business greater choice and better chances of finding the right products and services that meet any given organization's requirements.

In the services arena, competition is a particularly important force – it allows users to receive what they want and need, at a reasonable price. In a monopoly situation users have no choice but to accept whatever is available. In all major telecoms, IT and other traditional service areas, monopolies are being or have been dissolved, in order to drive competition and hence bring about an improvement in service levels. As an example, the previously state-owned global telecoms industry is being privatized, with the result that service levels increase, prices drop and, ultimately, the consumer benefits. It is also interesting to note that the usually large previously state owned telecoms enterprises are generally slower to react to market forces, customer demands and subsequently to implement change. This is thought to be due to the fact that these organizations are not used to acting with the customer in mind.

This document makes the assumption that competition and choice is of unconditional benefit to any given organization. Furthermore, makes the assumption that in order to truly operate in a non-monopolistic environment, Open Source Software must be viewed as a real, workable alternative to today's options. This reasoning is based on the following points:

- Different suppliers can supply the same product, since all specifications, documentation and source-code is freely and widely available to everyone;
- No patents have been, or can ever be, taken out on the software, and there are no restrictions on the use of the software by any party.²

Research and live, in-the-field testing has proven that Open Source Software is a viable technology alternative to today's proprietary software offerings.

Licensing

Software Licensing is an extremely complex topic in itself, and has assured many lawyers of a good pension plan. This section does not attempt to discuss all the nuances of software licensing, but rather highlights some current licensing issues that organizations are exposed to today, and the risks associated with those licenses.

A software license is in essence a contract between the creator/supplier of the software, and the user that will use the software. One hardly ever buys software, one buys a license. This is an important distinction, in the sense that buying the software means that the software itself becomes property of the buyer. Under current licensing practice, some organizations

² There are some restrictions on redistribution of Open Source software according to the licenses governing the software (e.g. the Gnu Public License). However, these restrictions do not restrict a business that intends to use the software, or modify the software for its own use, rather than redistributing it.

don't even own the CDs, media or manuals that come with some products. The bulk of the current licenses/contracts that exist between organizations and most commercial software vendors have some common points:

- The license usually confers the limited right of use of the software onto the user.
- The software itself is not property of the organization.
- Additional licensing terms may be added in the future. Refusal on the organization's part to agree with these additional conditions usually lead to termination of the complete agreement.
- The license may be withdrawn at any moment under certain conditions.
- The software may be subject to 3rd party license conditions that do not have to be made known in advance to the buyer, but may affect the buyer. This is usually due to foreign Governmental restrictions, as is the case in export of encryption technologies from Countries that are signatory to the "Wassenaar Agreement", or can also be due to (often unknown) incorporation of 3rd party technologies.

This last point has the potential to expose an organization to industrial or Governmental espionage, as was discovered by the Swedish Government in 1997, when it was found that Lotus Notes – their groupware network – had a specific feature called a "workfactor reduction field" that is built into Notes and incorporated into all email sent by non-US users of the system. The feature reportedly broadcasts 24 of the 64 bits of the key used for each communication, and relies on a public key that can only be read by the NSA³.

While these terms seem draconian, they are contract terms that an organization has already agreed to. Moreover, current US and EU legislation, as well as some instances of case law, have upheld some of these terms in the courts. While the EU is "wising up" to these issues, and is placing specific limitations on software licensing practices, the US is more libertarian in its approach.

The devil is in the details, and this was never more true than with software licensing. Besides the issues highlighted in the previous list, individual packages are governed by their own End User License Agreement (EULA). Some of these licenses have interesting implications to say the least.

While some of the licensing terms discussed may or may not stand up in a US or European Court, the real issue here is as follows: according to the terms of the contract between the customer and the vendor, the vendor retains all rights to the software, including the right to let the customer use the software. This right may be revoked at any time for any piece of software written after the passing of the UCITA Laws in the US, and similar laws in the EU. Some examples:

3 <http://www.wired.com/news/politics/0,1283,19602-2,00.html>

Microsoft FrontPage 2002 (Frontpage XP)

MS FrontPage is a website creation toolkit. It allows user to create webpages and websites. As is generally known, the web is haven for freedom of speech. It is therefore with some surprise that the following clause has found its way into the contract that governs the use of FrontPage:

"You may not use the software in connection with any site that disparages Microsoft, MSN, MSNBC, Expedia, or their products or services, infringe any intellectual property or other rights of these parties, violate any state federal or international law, or promote racism, hatred, or pornography."⁴

There are many different types of issues with this particular license. While the restrictions on free speech are clearly indefensible, and the lack of definitions for racism, pornography and hatred are simply bad legal decisions, the main impact on any organization using Frontpage is that Frontpage may not be used to disparage Microsoft, or its affiliates. This could be read to mean that Frontpage may not be used to create a website that details the failings of Microsoft software, business practices or any other type of criticism. Also, an organization's internal project site that compares competing software to Microsoft's could be illegal.

These type of clauses, while on the one hand are simply laughable, on the other hand are pretty serious. This is evidence of a clear, formalized attempt of Microsoft to control freedom of speech, and mode of expression. An organization needs to decide whether contracts carrying these type of clauses place undesirable restrictions on that organization.

Microsoft XML Licensing

XML is without a doubt the core technology for future exchange of information of all types. Already today, a significant amount of information available in electronic form is formatted and described in XML. XML is not the data itself – it is a way of describing data that makes the data meaningful and useful, in the same way that pages and a cover make a book useful, or a record or CD make music useful. However, the concept of XML goes further than that. Besides “packaging” information, it also describes *how* the information should be handled, and -- in some cases -- what it means. It clear that XML is a seriously powerful technology, bound to deliver hitherto unknown levels of interoperability between disparate systems and platforms.

This enviable position would not be attainable were it not for the fact that XML is an *Open* standard. i.e. The way the XML language works and the way it should be used is freely available to all. If XML were to be proprietary, it would be just another proposal, from just another vendor, albeit a very good one. The strength of XML lies in the fact that it is open and cross-platform, as opposed to proprietary.

Microsoft has a different opinion, however, and believes it should own major parts of the XML format. While this might seem an unattainable goal, since the XML format is based on an Open standard, the dominance of Microsoft on both the desktop, and to a lesser degree the server, combined with some craftily worded licensing agreements might make this possible. A particular Microsoft license agreement covering certain use of XML states:

"... You are not licensed to use the DOCUMENTATION to develop or assist in the development of (a) general purpose word processing, spreadsheet and/or presentation product(s) or an integrated work(s) or product suite(s) whose components include a general purpose word processing, spreadsheet and/or presentation product, or (b) a software program or filter that can convert or assist in the conversion of a file authored in the format specified in the DOCUMENTATION into another file format. ..."⁵

4 See Microsoft Frontpage 2002 License agreement.

5 See GartnerGroup T-11-6716

While one tends to think that the fact that Office 2000 and greater can save and open files in HTML and XML format would do miracles for interoperability, this license puts rest to this particular Microsoft marketing myth. It is clearly illegal to create anything that allows another (non-Microsoft product) to read and write Microsoft Office files.

This has a *significant* impact on any organizations business continuity. If, for any reason whatsoever, an organization wishes to migrate to a non-Microsoft Office suite in the future, required document conversions – already a difficult task – might be made illegal through this type of licensing.

The issue doesn't stop there however. Most of Microsoft's back-end infrastructure, such as Biztalk, SQLServer, Exchange and Commerce Server utilize XML as their core data interchange format, both internally as well as for interoperability purposes. The propensity that is shown for this type of restrictive licenses will probably lead to situations where it would be illegal to, for example, pump data from SQLServer to Informix in it's usable format (XML). Whereas the IT community as a whole sees XML as a liberating standard that will allow free interoperability, Microsoft sees it as an opportunity to rope in more customers.

This is not a "vague prediction", but already a reality. Both the above portion of the license, as well as Microsoft's current practices with respect to the LDAP, Kerberos, DNS and many other standard protocols have already created legal situations whereby the customer is allowed to inter-operate in order to migrate from a given platform to the Microsoft platform, as opposed to truly integrating disparate platforms.

Steve Ballmer, Microsoft's current CEO, said at the Gartner Symposium 2000:

*"In adopting Internet standards such as XML (Extensible Markup Language) as part of its .Net initiative, Microsoft will continue to protect any intellectual property that it embeds as objects in XML wrappers. "We will have proprietary formats to protect our intellectual property," he said. At the same time, Microsoft is committed to "a certain level of interoperability," and it is committed to standard protocols."*⁶

The risks faced by any given organization today in creating documents in, and migrating documents to, Microsoft's XML format, as well the use of Microsoft XML formats for the interchange of data are severe. There exists a real threat for the loss of ownership on *data in a usable format*. i.e. While the data as owned by an organization cannot ever become anybody else's property, the format in which this data is stored **is** property of another entity.

The data in itself has little value without the means of being able to use this data freely.

Microsoft Software Developers Kit (Internet SDK)

Another interesting snippet of a contract is from the Mobile Internet SDK. This toolset is used to create web-based applications:

"(c) Open Source. Recipient's license rights to the Software are conditioned upon Recipient (i) not distributing such Software, in whole or in part, in conjunction with Potentially Viral Software (as defined below); and (ii) not using Potentially Viral Software (e.g. tools) to develop Recipient software which includes the Software, in whole or in part. For purposes of the foregoing, "Potentially Viral Software" means software which is licensed pursuant to terms that: (x) create, or purport to create, obligations for Microsoft with respect to the Software or (y) grant, or purport to grant, to any third party any rights to or immunities under Microsoft's intellectual property or proprietary rights in the Software.

6 See <http://www.zdnet.com/zdnn/stories/news/0,4586,2642030,00.html>

By way of example but not limitation of the foregoing, Recipient shall not distribute the Software, in whole or in part, in conjunction with any Publicly Available Software.

"Publicly Available Software" means each of (i) any software that contains, or is derived in any manner (in whole or in part) from, any software that is distributed as free software, open source software (e.g. Linux) or similar licensing or distribution models; and (ii) any software that requires as a condition of use, modification and/or distribution of such software that other software distributed with such software (A) be disclosed or distributed in source code form; (B) be licensed for the purpose of making derivative works; or (C) be redistributable at no charge. Publicly Available Software includes, without limitation, software licensed or distributed under any of the following licenses or distribution models, or licenses or distribution models similar to any of the following: (A) GNU's General Public License (GPL) or Lesser/Library GPL (LGPL), (B) The Artistic License (e.g., PERL), (C) the Mozilla Public License, (D) the Netscape Public License, (E) the Sun Community Source License (SCSL), and (F) the Sun Industry Standards License (SISL)."

With this license, Microsoft clearly prohibits the use of Open Source software in connection with its own software. Since this applies to web-based applications, it is illegal to serve these applications from, say, the Apache web-server. The Apache web-server currently has a 60% market share on the world's web-servers, as opposed to Microsoft's 25%. Also, authentication, using OpenLDAP for example would be illegal, as would be the serving of the domain-name using the industry standard BIND DNS software. If the application needs to interoperate with Sendmail, for example, that also would be prohibited.

This however, does not make that application unusable – One could always opt to utilize Microsoft Internet Information Server, Microsoft DNS Server, Microsoft Active Directory and the Microsoft Exchange mail server.

The use of Orwellian “doublespeak” (“Viral Software”), while not relevant to this document, is interesting. This “Viral Software” has consistently outperformed Microsoft variants in market share, performance and stability – so as “Viral Software” it has been extremely successful.

These are some of the more “interesting” examples software licensing terms. However, almost every other proprietary software package created by Microsoft or several other software companies, such as Intuit, Network Associates, or Adobe has similar terms and conditions.

Office Automation Suite

Why would a change in office automation suite be required? Microsoft Office is without a doubt the market leader, with a significant amount of users world-wide. MS Office skills are ubiquitous, and according to the Microsoft marketing message, MS Office improves with every version. This section of this document will evaluate the state-of-play within the office automation suite market place, the costs associated with usage of Office, and several other issues.

There are today many office automation suites available, and even more stand-alone products. Office suites are preferred over collections of standalone products due to the better integration and similarity of interfaces offered by office suites. This is in line with the previously stated goals. The office suites presented in this document share some commonalities: they are all perceived as being part of the leading segment of the market, they share a common base functionality level, all support some form of automation, and all have high quality support structures available from the vendor.

Microsoft Office (all versions)

The purpose of this document, and in particular this section, is to have a technical head-to-head between different products. However, some comparison must be made between applications in order to provide contrast. As already mentioned, MS Office is by far the largest selling suite on the market today. The rise of MS Office is a spectacular success story.

In the “good old days of DOS” the top applications to use for office automation were Lotus 1-2-3 for spreadsheet work, and WordPerfect. Mastering these applications, without the GUI as we know it today, was something of a black art, and required many frustrating hours of dedication, and frustration. Once the applications were mastered by the end-user, the user was naturally loath to have to go through this grueling process again – anything else had to offer a significant improvement over the toolset in use.

The biggest shortcoming of both Lotus 1-2-3, as well as WordPerfect were the lack of What-You-See-Is-What-You-Get (WYSIWIG) feedback – the stuff displayed on the screen had absolutely no connection to what would eventually roll out of the printer. In fact many office “guru” made him/herself an irreplaceable asset to an organization, simply due to the ability to coerce the printer to produce usable printouts.

Microsoft was quick to recognize this shortcoming, and produced Microsoft Excel – the first spreadsheet application to feature a GUI and an almost overnight hit. Several key tactics were deployed by Microsoft to convince the user to give up Lotus 1-2-3. Complicated keystrokes, including menu pop-ups were “borrowed” from Lotus 1-2-3, allowing users long used to doing things the Lotus way to continue their work in the way to which they were accustomed. Almost flawless import capabilities ensured that the transition from Lotus 1-2-3 to Excel was virtually painless. Most users threw away Lotus 1-2-3 with glee and never looked back. Microsoft immediately repeated this action with Word vs. WordPerfect – employing the same tactics, and additional lessons learned from the Lotus/Excel switch. Shortly thereafter, PowerPoint followed to take the limelight away from Harvard Graphics as the presentation tool of choice, and Microsoft was the absolute winner in the office suite arena.

The remarkable thing about this whole switch is that the WYSIWIG feature is generally regarded as the *only* reason users switched – it was a tangible innovation, delivering immediate, tangible results. In some ways, the MS Office suite was inferior from a functionality perspective to the products that the users left behind. However, the WYSIWIG features were enough to persuade the user base to a massive switch.

The vendors of the other products were taken by surprise – this was a move that was not expected, and they had no ready answer to the threat from Microsoft. GUI's were generally regarded as a fad – the playing ground of a few graphic designers, and they were all using Macs, anyway – the real business user used DOS on a PC.

Microsoft rightly saw the need in the market and capitalized on that need. The rest, as they say, is history.

Vendors scrambled to deliver graphical versions of their applications, but the damage was already done. From there onwards, Microsoft continued building on the office suite. Microsoft Office is the cornerstone of the Microsoft empire – neither any of the Windows versions, nor any other application have delivered such a strong contribution to the continued success of Microsoft. Often, Windows is fingered as Microsoft's success story – however, this is incorrect. As a GUI, Windows 3.1, as well as Windows 95, suffered from some serious design flaws, both from a usability perspective as well as from a computer architectural perspective. Their success is mainly due to the initial success of Microsoft Office.

When discussing different office suites, the frequent comparison is between Microsoft Office vs. everything else. This view was the correct one to take during the days of the switch from text based to GUI based office automation. Since today the differences between office applications are nearly non-existent, this type of comparison can no longer be made.

Comparing any other suite to Microsoft Office is an understandable practice, however this implies that Microsoft Office *is a 100% representation of the business requirements*. This is obviously not the case – the fact that a user gets work done with a tool does not mean that the tool is a perfect match to business requirements (it helps, though). As a case in point, Excel is an extremely powerful spreadsheet solution – however, the average user might only use about 20% of the functionality available. However, different users use a different 20%. This goes for all other applications within the office suite. This means that perhaps 80% of the application sitting on the user's desktop is wasted. Additionally, it would be difficult to find any heavy user of any MS Office application who couldn't name at least one feature they would like to have which does not exist in that application.

While this implies that at the end of the day all users' requirements are met, it does not mean that all business requirements are met – on the contrary, this approach conflicts with the business requirements in some critical areas. This is an area that will be touched on later in this document. The issue is that the point of departure for selecting an Office suite should *not* be the “solution du-jour”, but should be the business requirements. The needs and wants of the end-user play an important role in the formulation of these requirements, but are not the only things that merit attention.

Having said that, the latest incarnation of Microsoft Office – Office XP – is an excellent product, but very expensive, in the same way that the Rolls Royce is an excellent vehicle, but beyond the means of most people. The expense is not only in terms of money, but also in terms of interoperability, licensing, choice and freedom.

Microsoft has paid a great deal of attention to the GUI – it looks both “hip” as well as “Internet”. The product is, on the whole, well designed, offers a rich user experience and, when used on a full Microsoft backend, good integration facilities. On the other hand, a lot of the standard features of the 2000 or XP Suite are not likely to be used by the mainstream of users, some of the features have clearly been shown to be counter-productive, and have been dropped from the latest versions.

Some other new “functionality” offered by Office XP has drawn wide criticism from users, analysts and CIO's. The new “Smart Tags” feature in Office XP will perform continuous “intelligent” scans on the document in question, and will automatically create a (proprietary) form of hyperlink as soon as it comes across a piece of familiar text. For example, when typing in the name of a City, this may link to a map of that city located on the Internet (of course, from the Microsoft Encarta website) or when typing in the name of a contact, Smart

Tags will automatically link to the contact information available on that individual (Provided Microsoft Outlook and/or Microsoft Exchange are used). When typing in the name of a brand or company, this will automatically link to the website of the organization (provided this organization has paid Microsoft for this service).

This feature has drawn criticism for several reasons, mainly the fact that only Microsoft, or organizations who pay a royalty to Microsoft seem to benefit from this feature – the implementation details for Smart Tags are vague, and access to the Smart Tag list seems to be prohibited. Moreover, Microsoft has been accused of simply broadening the audience for targeted advertisements, using Smart Tags not as an end-user enhancement, but rather as an advertisement tool.

Another new feature of Office XP is the “activation wizard”. This marvel of modern technology prohibits use of Office on a machine until it has connected to the Microsoft servers, via the Internet, and registered on-line. This wizard requires user intervention during installation, the entering of a complex code, and assumes a live Internet connection. Moreover, it has been found under certain circumstances to transmit information to Microsoft regarding machine configuration, other software installed and other unique identification data. If the machine where Office XP is installed should require a significant change in hardware configuration, or requires to be restored from backup after a crash, the activation wizard kicks in, and the product can only be activated after placing a phone call to Microsoft.

The activation wizard is wildly unpopular among IT professionals – first off, it has already failed in its original goal (curbing of piracy) since the codes and routines were hacked even before the product hit the market. Secondly it places an excessive burden on systems administration staff, who must be present at installation, defeating any automated roll-out mechanism. To combat this last issue, Microsoft makes available a version of Office XP that does not use the activation wizard – for organizations that use the Software Assurance licensing scheme or have another type of Enterprise Agreement with Microsoft. A multitude of research by several different analysts, such as the GartnerGroup, has shown that this licensing scheme is rarely in the interest of the end consumer.

Lotus SmartSuite

IBM/Lotus offers the Lotus SmartSuite Office productivity applications. Lotus SmartSuite is a run-of-the-mill office productivity suite, that offers particularly good integration with a Lotus Notes based back-end infrastructure. However, SmartSuite suffers from the same handicap as Lotus Notes: a counter-intuitive GUI. The GUI offered by Lotus products is completely divorced from what users expect, and users often report frustration while trying to get used to the new interface. Finally, IBM has admitted that it views SmartSuite as a “mature” product and will no longer invest in it.

Corel WordPerfect Suite

The Corel WordPerfect suite is especially popular in environments where users need to create large, and complexly formatted documents. These include legal, educational and Governmental organizations. While it offers the same type of functionalities as other mainstream office automation suites, it has the added benefit of being able to reveal formatting codes within the document – a feature much loved by those who have been able to master their usage.

While WordPerfect users are quick to tout the superiority of WordPerfect over MS Word, they tend to be a lot more quiet about the other parts of the office suite. In particular, Quattro Pro (Corel's spreadsheet application) is widely considered inferior in features and ease of use to MS Excel, a significant weakness in the office suite market. Companies that favor WordPerfect often end up in the unenviable position of trying to economically license WordPerfect and MS Excel on most user's desktops.

Microsoft has recently acquired a significant stake in Corel, Corel is suffering from severe financial troubles and Corel's only leader has packed up and left. While The cost of purchase for Corel is about a third of the cost of purchase for Microsoft Office, the previously discussed issues with other components of the Corel Office Suite, and associated difficulties in implementation offers little incentive to switch to Corel. In general, the Corel solution is adopted by those organizations that still use Wordperfect.

StarOffice Family

In 1999, Sun Microsystems acquired the StarOffice Suite from the German "Star Division" company. Sun immediately offered StarOffice for free download from its website – a move that has created considerable ripples within the software industry. StarOffice is a feature-complete advanced office automation suite, offering excellent file and feature compatibility with most other mainstream office automation suites. StarOffice is both multi-platform as well as multi-user, and is particularly well suited for use in a centralized environment.

The latest production version, StarOffice 5.2, was far from perfect however. Some interface features were much disliked by its users: for example, StarOffice 5.2 placed it's own version of a desktop environment over the environment in use – something that had a lot of end-users confused. Also, StarOffice was notoriously slow – some sites reported initial startup times of over 120 seconds, and document loading of complex documents could, in some cases, take over 5 minutes!

Sun Microsystems – always known to be a strong supporter of the Open Source community – decided in the beginning of 2000 to open the source code under an Open Source license, under the name OpenOffice.org. This move seriously changed the competitive landscape of the office automation suite market. A global developers community had -- for the first time – access to almost the full sourcecode of an office suite. Thousands of developers and users from all over the world had a chance not only to work on a product of significant magnitude, but also had the opportunity to directly influence the feature set and architecture of the product.

During the course of the past year, this has lead to some dramatic improvements. Several unpopular features, such as the previously mentioned desktop, but also a web-browser and email component, were removed from the office suite, loading times and response times have seen improvements of over 200%, and the whole suite has become easier to use, as well as more streamlined as the previous version.

Moreover, the whole of OpenOffice.org has become compliant with open standards where applicable. As an example, the file format of OpenOffice.org files is Open Source, free, and widely available. Like the new Microsoft Office filestandard, it is based on XML. Unlike the Microsoft format, it is non-proprietary, well documented and freely available. The OpenOffice.org team is currently working together with other Open Source office suite teams, such as Koffice⁸, to create a *common* format.

If this format were to materialize, this could spell the end of proprietary standards – a tool used by software makers in a dominant market position to keep customers locked in to their solutions. Since the Open Source movement does not rely on these methods to keep customers, but rather believes that superior technology and good match with business requirements should keep customers loyal, the creation of a common, sensible file format only makes sense.

The development process of OpenSource software is markedly different from commercial software development, and Sun, together with Collab.net have managed the transition of developer base remarkably well. Currently, there are three main products created off the OpenOffice.org codebase, and, due to the nature of OpenSource development, anybody is free to create their own version. The following sections enter into the differences between the

⁸ KOffice is not compared in this document. While it is a great package, with much potential, it still lacks critical features that are required for consideration on the corporate desktop.

versions of OpenOffice.org currently available.

OpenOffice.org

OpenOffice.org build 638C is the main codebase for all current and future OpenOffice/StarOffice versions. It is fully Open Source, licensed under the Sun Industry Standards Software License (SISSL) and the Lesser GPL (LGPL), and is Compliant with the OpenSource licensing standard as per the OSI approved licenses⁹

Sun StarOffice

Sun StarOffice 6.0 Beta 1 is exactly the same as the OpenOffice 638C codebase, with the exception of support for Asian Fonts and input methods and a few other features. These pieces of software are proprietary and licensed from 3rd parties, and Sun is therefore not allowed to re-release this code under an OpenSource license. The software is still, however, free-of-charge. Sun's primary means of deriving revenue from StarOffice is through the offering of training, support and consultancy services, although Sun is considering charging a nominal fee for the final version of the software. Also, Sun hopes to increase hardware sales (their core business) through the provision of integrated hardware/software solutions. This strategy has been successfully employed by Sun in the past.

Sun ONE Webtop

The Sun Open Network Environment (ONE) Webtop is not really a product, it is more of an architectural concept. However, one usable end-product available today through the ONE architecture is the Sun Webtop. The Sun Webtop is a selection of OpenOffice.org code, compiled as separate components, and integrated into the ONE environment. This means that the Sun Webtop offers a fully browser based office productivity platform. Not only is the suite multi-user, and multi-platform – it is also multi-device.

The Sun Webtop allows users to simply use a browser for all their office productivity needs – documents can be viewed, and edited, using almost any device. The ONE infrastructure – if build correctly – should be able to distinguish between different devices, so if one were to access the portal using a browser on a thin-client, or laptop, the “fat” application suite would be run – but if one were to use a PDA, a very small Java Applet would be run. This delivers a tremendous amount of design freedom to the enterprise. Wireless computing, wearable computing, and other type of devices could be more readily integrated into the core environment without design compromise to the offered functionality on such devices.

Overall, in terms of cost, flexibility, feature set, standards adherence and suitability, the different options based on OpenOffice.org forms a considerable contender for inclusion on any organizations approved and supported software list.

9 http://www.opensource.org/site_index.html

Competitive Selection and Requirements Match

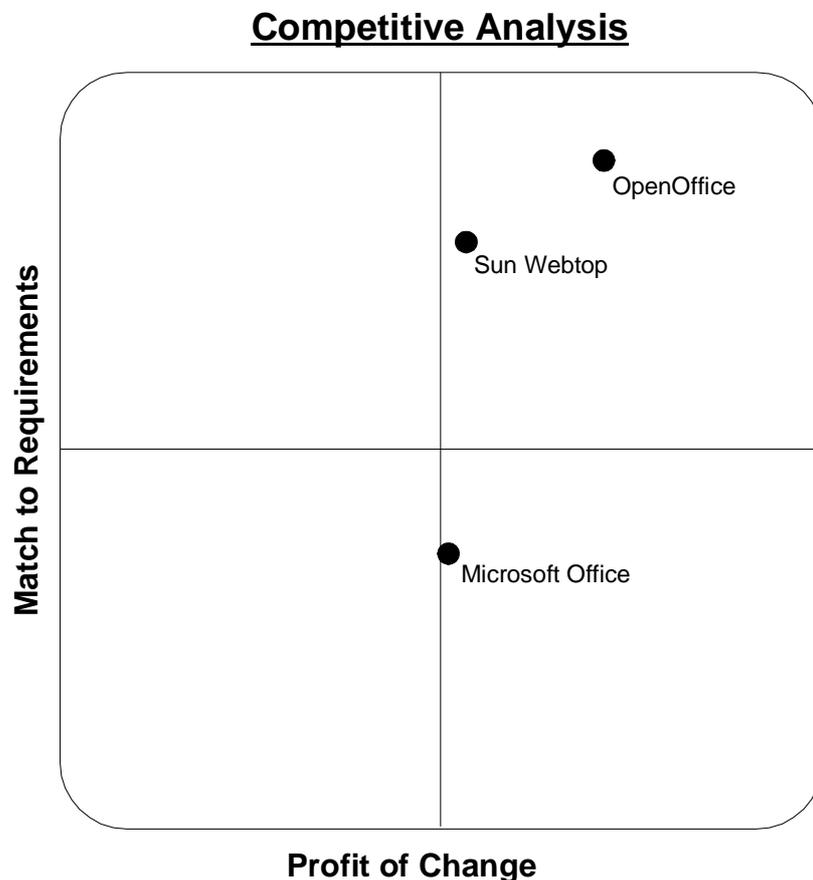
In order to determine what office suite makes the best fit for inclusion into an organizations approved software list, a scorecard based methodology is used. The scorecard is build on the assumption that all applications to be scored offer roughly the same level of functionality.

The packages to be tested against each other are Microsoft Office, OpenOffice.org, and the Sun Webtop. The testing criteria are based on the following:

- The package must mitigate the issues in the problem statement;
- The package must meet the goals as set forth in this document;
- The package should deliver the highest possible profit of change;

All packages have been assessed against these criteria. An analysis of how each package performed against these criteria is presented.

Chart 1 - Office Suite Competitive Analysis



Microsoft Office does not mitigate the issues laid out in the problem statement. In fact, Microsoft Office is seen to exacerbate these issues. This is due to the fact that a large portion of the problems are created by Microsoft's licensing issues, with several significant licensing issues with the Microsoft Office XML file format and FrontPage freedom of speech restrictions. Sun Webtop suffers less from licensing issues, but the heavy dependence on Java brings licensing issues of its own – albeit not a severe as those introduced by MS Office. OpenOffice, with an OpenSource License, has no licensing issues at all, since the various OpenSource licensing schemes all strike a sensible balance between commercial interests and consumer interest.

In the area of costs, Microsoft Office comes at a high cost of purchase – especially due to Microsoft's recent changes in their volume licensing program. Sun Webtop comes at a lower cost, but requires a complex infrastructure. OpenOffice.org is completely free of charge, in any configuration, so OpenOffice scores highest.

In terms of Vendor Independence, Microsoft Office again fails to impress. Microsoft has employed several artificial technology constraints to limit the consumer's choice in terms of operating system, file compatibility and support. Sun's Webtop, while offering a significant amount of freedom in terms of deployment platform, still limits the user to using Java and the Sun ONE architecture. OpenOffice.org places no such constraints – thus offering the highest vendor independence.

In terms of desktop management costs, Microsoft Office and OpenOffice.org are matched. Both applications have reasonable, though not perfect technologies in place to tame the costs of managing the desktop environment. Even though OpenOffice.org allows for out-of-the-box multi-user deployment, thus working toward lower desktop Management costs, Microsoft Office has a mature policy and profile management system in place. Sun Webtop scores highest – a pure web-based application only requires a browser at the desktop – significantly lowering costs. In terms of general configuration management costs however, Microsoft Office again scores the lowest – the management of MS Office requires a lot of expert staff and time, whereas OpenOffice.org has significantly simplified management requirements. Again, Sun Webtop's centralized, browser based architecture delivers high points – configuration management issues are non-existent on this model.

In terms of goal matching, Microsoft Office does not match well with the developer efficiency directive. This is a difficult issue to score, since all Microsoft platforms and applications deliver a uniform development model. While this may sound like a good thing, and this often is a good thing, the truth is in the details. The Microsoft development model is totally controlled by Microsoft, and often abused to further the business goals of Microsoft. The development environment delivered by Microsoft, while powerful and advanced, is also very proprietary, and actively bars all non-Microsoft based products and platforms from the available options. A solid, efficient development environment is based on open standards, as opposed to proprietary models – something that cannot be said for the Microsoft model.

Moreover, while Microsoft Office offers a consistent development environment across both the Office Automation Suite, as well as the various Windows Operating Systems, Microsoft's failure to adhere to open standards, combined with the absence of support for other environments such as Linux, creates unacceptable barriers to those developers and organizations that need or want to support heterogeneous systems. This is not in line with the “resource sharing” directive as outlined in the “Vision and Goals” section.

Looking at improving portability and scalability again delivers a low score for Microsoft. Portability is non-existent on Microsoft Office – it will only run on Microsoft Windows. Scalability is also very difficult to obtain – since it is a fat-client only application. Both OpenOffice.org as well as Sun Webtop are highly portable (they are multi-platform) as well as highly scalable, with OpenOffice.org winning an extra point over Webtop, since Webtop can be scaled well, but the upscaling of Webtop is more complex.

Improving interoperability delivers high marks to OpenOffice.org as well as Webtop – they both adhere to open standards, are transparent, have open data models and expose solid interfaces.

Vendor independence. The non-proprietary nature of OpenOffice.org give the suite an easy win.

When looking at the reduction of life-cycle costs, the high cost associated with purchasing and operating Microsoft Office do not render it the most attractive of deals. The TCO associated with both OpenOffice.org as well as Webtop is very low, due to several factors already discussed.

The security aspects of the evaluated packages are significantly different. The UNIX/Linux platforms are known to be very secure when compared to the Microsoft platform – Microsoft has, for example, long suffered from document based viruses. Due to the superior security models of both UNIX and Linux, document viruses pose a low risk to OpenOffice.org and Webtop. Microsoft on the other hand has a long and rich track record when it comes to security, none of it positive.

Manageability is another key aspect that is evaluated. Here Microsoft Office manages to score higher marks – this is due to the fact that Microsoft has had the time and feedback to sort out the manageability issues surrounding Office – however, the superior pedigree of the UNIX/ and Linux platforms ensure that both OpenOffice.org as well as Webtop are close behind.

When looking at TCO, OpenOffice.org offers the lowest TCO, when deployed as a multi-user, server based application. In terms of productivity, all applications score roughly equally. Microsoft has invested a lot of money into basic usability research, and this is openly reflected in the interface of Microsoft Office – an interface that has been closely mimicked by both OpenOffice.org as well as by Sun Webtop. Microsoft does deserve some extra credit for increasing user efficiency in general terms, however the overall product stability leaves much to be desired.

GNU Free Document License

GNU Free Documentation License
Version 1.1, March 2000

Copyright © 2000 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other written document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you".

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (For example, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, whose contents can be viewed and edited directly and straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and

that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup has been designed to thwart or discourage subsequent modification by readers is not Transparent. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML designed for human modification. Opaque formats include

PostScript, PDF, proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies of the Document numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a publicly-accessible computer-network location containing a complete Transparent copy of the Document, free of added material, which the general network-using public has access to download anonymously at no charge using public-standard network protocols. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has less than five).

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section entitled "History", and its title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. In any section entitled "Acknowledgements" or "Dedications", preserve the section's title, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section as "Endorsements" or to conflict in title with any Invariant Section.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections entitled "History" in the various original documents, forming one section entitled "History"; likewise combine any sections entitled "Acknowledgements", and any sections entitled "Dedications". You must delete all sections entitled "Endorsements."

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, does not as a whole count as a Modified Version of the Document, provided no compilation copyright is claimed for the compilation. Such a compilation is called an "aggregate", and this License does not apply to the other self-contained works thus compiled with the Document, on account of their being thus compiled, if they are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one quarter of the entire aggregate, the Document's Cover Texts may be placed on covers that surround only the Document within the aggregate. Otherwise they must appear on covers around the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License provided that you also include the original English version of this License. In case of a disagreement between the translation and the original English version of this License, the original English version will prevail.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.