

Open Document Format (ODF)

A Guidance for Governments

Paolo Dongilli - Autonomous Province of Bozen-Bolzano

v0.1 - 28 October 2015

Abstract

The UK government has selected ODF 1.2 as the standard for editable office documents to be used across government. ODF 1.2 was selected as the standard for government because it (a) allows citizens, businesses and other organisations to share and edit documents with government - and the other way round (b) allows people working in government to share and edit documents with each other (c) is compatible with a wide range of software (d) is a reliable long-term solution for storing and accessing information. This slightly modified version of the original guidance, gives general information on the standard, as well as more detailed information for chief technology officers and procurement officers of all governments that want to avoid lock-in by using the world's leading open document standard.

Contents

1	Foreword	5
1.1	Copyright Notice and Licensing Terms	5
1.2	Acknowledgements	6
2	Introduction to Open Document Format (ODF)	6
2.1	ODF's main properties and features	6
2.2	How ODF works	6
2.3	Reasons your organisation should use ODF	7
2.3.1	ODF allows you to work with a wide range of tools and devices	7
2.3.2	ODF is flexible, versatile and lowers IT costs	8
2.3.3	ODF allows you to store and access information over the long term	8
2.3.4	ODF keeps your organisation's data more secure	9
2.4	Who maintains and develops ODF	9
3	Procure ODF solutions	10
3.1	Buy products and services based on ODF	10
3.2	ODF requirements for tendering	10
3.2.1	Core requirements	11

3.2.2	Full support for ODF	11
3.2.3	Solution-specific extensions	11
3.2.4	Cross-platform use	11
3.3	Supplier confirmation of ODF support	12
3.4	Using free open source software based on ODF	12
4	Base ODF solutions on user needs	12
4.1	Plan your user research	13
4.1.1	Identify user types	13
4.2	Questions for users: text documents, spreadsheets, presentations, databases	14
4.2.1	Text documents	15
4.2.2	Spreadsheets	15
4.2.3	Presentation documents	16
4.2.4	Databases	16
4.3	Select a solution based on user needs	17
4.3.1	Create a shortlist	17
4.3.2	Prioritise functionality	17
4.3.3	Assess	18
4.3.4	Aim for a baseline service	18
4.3.5	Identify suitable combinations	18
4.3.6	Procurement phase	18
4.3.7	Users outside your organisation	19
5	Validators and compliance testing	19
5.1	Reasons for compliance	19
5.2	Use validators to test compliance	20
5.2.1	Which validator to use	20
5.2.2	Interpret the output of a validator	20
5.3	Use comparison software to test compliance	21
6	Platforms and devices	21
6.1	Operating systems	21
6.2	Web-based solutions	22
6.3	Security on platforms and devices	23
6.3.1	Connectivity requests	23
6.4	Accessibility on devices with alternative input	24
6.5	Developing platform or device-specific solutions	24
7	Accessibility	25
7.1	The accessibility requirement	25
7.2	Make documents more accessible	25
7.3	Product-specific support	26
7.4	ODF support of alternative media	26
7.5	Build solutions that implement accessibility	27
7.6	ODF accessibility guidelines for developers	27
8	Privacy and security	27
8.1	Minimise risks to the security of documents	28
8.2	Document encryption	28

8.3	Protection locks	29
8.4	Digital signatures	29
8.5	Save with password	30
8.6	Choose a good password	30
8.7	Work safely in the cloud	31
9	Avoid macros in documents	31
9.1	What macros are	31
9.2	Alternatives to macros	31
9.3	Assess your organisation's macros	32
9.4	Plan your move away from macros	32
9.5	Deal with macros post-migration	32
10	Integrate ODF with enterprise tools	32
10.1	Enterprise tools	32
10.2	Choose a platform	33
10.3	Automate tasks and processes	33
10.4	Records and documents management	34
10.5	Use metadata to record a document's history	34
10.6	Create new ODF templates	34
10.7	Replace legacy applications incompatible with ODF	35
10.8	Don't use direct formatting	35
10.9	Variable data printing	35
11	Extensions, plugins and custom solutions	36
11.1	Additional software options	36
11.2	Conforming documents	36
11.3	Image or graphic plugins	36
11.4	Validate output	37
12	Collaborate on documents	37
12.1	Applications that support collaboration	37
12.1.1	Web applications for collaboration	38
12.2	Collaborate offline	38
12.2.1	Annotations	39
12.2.2	Change tracking	39
12.2.3	Pre-publication tools	39
12.3	Collaborate in real time	39
12.3.1	Peer-to-peer (p2p)	40
12.3.2	Server-based with local hosting	40
12.3.3	Service-based, hosted by a third party	40
12.4	Bespoke solutions	41
13	Change tracking in ODF	41
13.1	Options that support change tracking	41
13.2	Risks and benefits	42
13.3	Automatically compare documents	42
13.4	Automated multiway merging	43
13.5	Alternatives	43
14	Embed fonts in ODF documents	43

14.1	Open licence fonts	43
14.2	Freeware and shareware fonts	44
14.3	Microsoft and Windows fonts	44
14.3.1	Check if you can embed Microsoft and Windows fonts . .	45
14.3.2	Alternative open fonts	45
14.4	Add a font manually	45
15	Corporate styles and templates	45
15.1	Styles and templates	45
15.1.1	Styles	46
15.1.2	Templates	46
15.1.3	Styles and templates in ODF	46
15.2	Create templates and styles	47
15.3	Keep templates up to date	47
15.4	Dynamic content in templates	48
15.4.1	Variables	48
15.4.2	Form functionality	48
16	ODF spreadsheets and formulas	49
16.1	OpenFormula for data manipulation	49
16.2	Levels of OpenFormula implementation	49
16.3	Problems converting spreadsheets to ODF	50
16.4	Known issues	50
16.4.1	Precedence issue	50
16.4.2	Decimal separators	51
16.4.3	Empty parameters	51
16.4.4	Workarounds to Excel AND/ OR bug	51
16.4.5	Intersection operators	51
17	Support and training	51
17.1	Train users of office productivity tools	52
17.1.1	Identify what your users need	52
17.1.2	Training if you replace your existing tools	52
17.1.3	Formal qualifications for users	52
17.2	Training and support for technology teams	53
17.2.1	Building your team's capability	53
17.2.2	Bringing in skills from outside	53
17.3	Support, service and development	54
17.3.1	Vendor-owned solutions	54
17.3.2	Open source solutions	54
17.4	Support for using developers with ODF	55
17.4.1	Open source	55
17.5	Helpful organisations, tools and events	56
17.6	Support on the development of the standard	56
18	Overview of productivity software	56
18.1	Auxiliary products and services	56
18.2	AbiWord	57
18.3	Apache OpenOffice	57
18.4	Bread	58

18.5	Calligra Suite	59
18.6	EtherCalc	59
18.7	EuroOffice Professional	60
18.8	Gnumeric	61
18.9	Google Apps	61
18.10	Kotype	62
18.11	LibreOffice	63
18.12	Microsoft Office	63
18.13	Owncloud Documents	64
18.14	OX Text	65
18.15	Wodo.TextEditor component	65
19	Costs and benefits of ODF	66
19.1	Benefits of moving to ODF	66
19.2	Cost savings by European governments	66
19.2.1	France	67
19.2.2	Italy	67
19.2.3	Spain	68
19.2.4	Germany	68
19.2.5	The Netherlands	68
19.2.6	Other municipalities	68

1 Foreword

The present document in its initial release doesn't bring anything more, in terms of content, to the original UK guidance acknowledged below.

The need of a fork was mainly meant to render the document in a textual format (Pandoc's Markdown) in order to be able to store it onto a public repository (GitHub) and easily keep track of versions and changes. The original guidance is available in HTML format only, accessible by means of copy and paste from the UK Government web pages.

I thought the document had to be a general guidance for any government interested in adopting ODF as the world's leading open document standard.

Generalizing it, rendering it into Markdown and committing it onto GitHub will allow people to join the effort to keep it up-to-date, correct typos, add new content and easily fork it e.g. for translation purposes or any other purpose.

For converting the source of this document from Markdown format into other formats (in particular ODF), I suggest [Pandoc](#), a universal free software document converter available under the GPLv3 license.

1.1 Copyright Notice and Licensing Terms

Copyright (C) 2015 Crown Copyright

Copyright (C) 2015 Paolo Dongilli

This document is licensed under the [Open Government Licence v3.0](#).

1.2 Acknowledgements

The current document is a slightly revised version of the document “Open Document Format (ODF): guidance for UK government” released by the Cabinet Office of the UK Government as of 11 September 2015. The original document can be found here <https://www.gov.uk/guidance/open-document-format-odf-guidance-for-uk-government>, licensed under the Open Government Licence v3.0.

2 Introduction to Open Document Format (ODF)

An overview of ODF, how it works and why your organisation should use it.

2.1 ODF’s main properties and features

Open document format (ODF) has been an [international standard](#) for software since 2005. Herein we suggest all governments, as the UK government does, to use ODF for creating editable documents.

ODF allows users to send, view and share office documents regardless of what software they have and what device (e.g. tablet, mobile, laptop) they are using. To do this, ODF puts different types of office documents into a single file format that covers: - text documents - spreadsheets - charts - graphical documents, such as drawings and presentations

ODF also provides other useful features, such as:

- a track changes mechanism (which will be extended to offer [real-time collaborative editing](#))
- the ability to add [digital signatures](#) to a document
- a powerful generic [metadata](#) system

ODF consists of a family of technical specifications put together by an international consortium called [OASIS](#). OASIS includes representatives from hundreds of large and small software companies as well as academics, government representatives, civil society representatives and other stakeholders.

The current version is [ODF 1.2](#) and this is the version the government has adopted.

2.2 How ODF works

Many users won’t notice a difference when they use files saved in ODF. ODF keeps the document’s content and layout information separate so that they can be processed independently of each other. Each type of object in the file is given its own place, and there’s a neat index of every subfile inside. This unlocks

powerful new capabilities for office applications that were not available in earlier file formats, such as embedded [Linked Data](#).

ODF is not tied to a single supplier and could be used by any supplier in any product. It also provides a flexible container that can embed almost any existing digital object and allow it to be used in an office document. Custom metadata (such as [provenance data](#), [copyright information](#) and other forms of Linked Data) can be added anywhere, to any element within a document.

This is similar to how the HTML standard allows you to reference just about anything, such as:

- a TIFF image file from a 30 year old scanner
- a plugin object from a third party supplier (eg Adobe Flash)
- a custom font used to style a page

HTML does not define any of these, but just has to point to the right place.

Similarly, ODF opens many possibilities for integrating with business processes and internal applications. It allows you to, for example:

- embed a live webpage inside a presentation
- include a complete music score within a text document
- put a zoomable 3D globe inside a spreadsheet to show where your organisation is buying the most products and services

ODF is built on another common standard called [XML \(Extensible Markup Language\)](#), a well-known web standard for structured data created by the [World Wide Web Consortium \(W3C\)](#). This has helped make a versatile standard that has been quickly adopted in the marketplace. ODF specifications reuse quite a few other W3C standards, such as [RDF metadata](#), [Synchronised Multimedia](#) and [digital signatures](#).

2.3 Reasons your organisation should use ODF

The ODF standard has been [adopted for use by UK government organisations](#). It's used by governments across the world and supports the functionality that most users expect in modern office productivity software.

2.3.1 ODF allows you to work with a wide range of tools and devices

In the past, IT departments didn't need to take into account users outside their organisation when they were procuring office applications. Documents were created in a single office application and would travel electronically to a single brand of printers in the print room. External users weren't affected by the choices IT departments made.

Now most organisations have to interact with many external users on a daily basis. Because of that, IT departments have to start with user needs (see section

4 - Base ODF solutions on user needs) when they tender a contract for a new supplier.

Users can now create, collaborate and interact on a wide variety of devices, platforms, software tools and services. They should also be able to access government information without having to buy expensive software. ODF's goal is to make information accessible to all, regardless of the device they are using or the software they have.

Some people and businesses may have a specific need for customised software and hardware. You can't give a sight impaired employee a tablet app to do their work at the service desk of a supplier. Likewise, a private law firm will have different needs and financial means compared to a charity or local court.

Users need software and technology to work together seamlessly, because the output from one organisation is the input of the next. ODF helps provide the interoperability that will allow everyone to choose the tool that best meets their needs.

2.3.2 ODF is flexible, versatile and lowers IT costs

ODF shares many of the traits that are common between web- and device-based applications. For example, many office applications offer access to online editing environments through a web browser. This shared technology:

- lowers the cost for implementing ODF
- creates an economy of scale
- increases the product's robustness This shared technology allows users to benefit from the availability of [Linked Data](#) and to become part of it. Combining ODF with other XML data means that the structured information inside can be reused or consumed in ODF.

As ODF isn't tied to a single supplier, different suppliers can offer additional solutions that should be compatible with your existing ODF applications. The potential for healthy competition can help keep costs down.

2.3.3 ODF allows you to store and access information over the long term

Many older office files are made up of long binary sequences. Numbers, words, formatting and other information you enter as you type and click are converted into long rows of zeroes and ones. These make no sense outside of the specific applications that created them, so that application is needed to read that information. Interpreting a single bit of this binary code incorrectly could cause the application reading that file to behave in the wrong way. And there's no easy way to tell if errors occurred.

Understanding files written in these legacy formats is challenging even for experts, including the software developers that have written applications that process them. Some file formats also have reached the end of their lifecycle and are no

longer maintained by their original creator, for instance the various versions of .doc, .xls, .ppt and .wpd. ODF is the universal, supplier-neutral successor to all of these legacy formats.

Beyond the user needs of today, governments also have to consider future needs. Users need to be able to access many types of documents in the long term, including records, birth certificates and legal documents such as permits about property ownership or contracts.

ODF is an advanced and future-proof format that can cater for these needs.

2.3.4 ODF keeps your organisation's data more secure

Using ODF can help make organisations much less vulnerable to targeted attacks compared to older binary (legacy) formats. This reduces the number of computers infected by viruses, spyware and adware. Legacy office files are among organisations' top 3 most common vulnerabilities to targeted attacks.

[German research in 2011](#) indicated that the effectiveness of antivirus applications tested against attacks carried out through legacy text document files was limited. Three out of 4 antivirus solutions scored a recognition of 20% or less.

Legacy file formats (such as .doc, .xls and .wpd) were created when resilience against cyber attacks wasn't a significant part of design requirements, as computers were typically offline. The proprietary nature of these formats means you can't run a machine processable set of validation rules (see section 5 - Validations and compliance testing) to check if a document contains something unusual.

Binary documents make it easy to hide active attack software as the software used in an attack will often consist of short streams of zeroes and ones.

Of course, ODF itself does not make insecure software safe. It can only make it easier to check exactly what is going in and coming out of the document. It is up to the individual application and anti-malware measures to protect you and the people you communicate with.

2.4 Who maintains and develops ODF

The [Open Document Format for Office Applications Technical Committee \(ODF TC\)](#) produces and maintains ODF.

Current [members of the ODF TC](#) include:

- publicly traded companies (such as IBM, Microsoft, Red Hat and Beijing Venustech), small and medium enterprises
- a number of open source communities (Multiracio, Collabora Productivity Ltd, The Document Foundation, KDE e.V.)
- independent industry experts and representatives of [ISO/IEC JTC1 SC34](#)

SC34 is a sub-committee of the [International Standardization Organization \(ISO\)](#) and [International Electrotechnical Commission \(IEC\)](#) committee.

OASIS is officially responsible for maintaining the ODF standard, but ODF is also officially published as an [International Standard ISO/IEC 26300](#) by ISO/IEC JTC 1.

Anyone can participate within OASIS directly or through membership of a national standards body such as the [BSI](#).

You can also send comments to a [public mailing list](#) maintained by the ODF TC members.

3 Procure ODF solutions

An overview of steps you should take when procuring applications or services dealing with editable documents, to ensure they comply with the ODF standard.

3.1 Buy products and services based on ODF

When procuring an application, product or service that deals with editable documents and their lifecycle you must make sure your specification includes ODF compliance. Editable documents include text documents, spreadsheets and presentations. A document's lifecycle involves creation, viewing, changing (including revisions), exchanging and archiving.

Documents that are for viewing rather than editing, e.g. PDFs, must also comply with a [separate set of open standards](#). Read about open standards and how they relate to procurement in the UK government's [Open Standards Principles](#).

Make sure the ODF requirement for services and products is clear in your tender documentation, particularly in the tender specification. Your contract should specify that the supplier is required to ensure ODF support at the time of deployment, according to the agreed specifications and requirements.

3.2 ODF requirements for tendering

Include a clear statement in your technical specification of your functional and user needs (see section 4 - Base ODF solutions on user needs) so that suppliers can make sure that the proposed solution meets them. Review the government policy paper on sharing or collaborating with government documents (an example from the UK can be found here <https://www.gov.uk/government/publications/open-standards-for-government/sharing-or-collaborating-with-government-documents> to help prepare your statement. You can also download and use [this sample text](#) for ODF procurement specifications.

3.2.1 Core requirements

Your procurement documentation must include the requirement that the application or service works reliably with revisable ODF text documents, spreadsheets and presentations for purposes of:

- generating
- importing
- viewing
- editing
- storing
- tracking changes (where required; specify that ODF documents keep track of changes to the document so they can be reviewed and individually approved or rejected)

3.2.2 Full support for ODF

You should also ensure that:

- products retain all relevant information when storing output in ODF or importing or exporting data
- there are no dependencies on other file formats
- suppliers don't hardwire their software with a specific office application but not support the standard

3.2.3 Solution-specific extensions

ODF is able to incorporate 'foreign elements'. These are data elements that can be added to give software developers some flexibility to offer added functionality. These foreign elements may make documents less portable (ie possibly incompatible with some other tools) and can make functionality more difficult to migrate from one solution to another. If a different application can't interpret the meaning of some elements within a document, it won't be able to make use of the information it contains.

You should understand where and how software makes use of such [foreign elements](#). You should state in the procurement documents that suppliers should provide complete documentation about any foreign elements used in their software. You may request sample documents to test in multiple applications, or ask suppliers to demonstrate compliance with an ODF validator (see section 5 - Validators and compliance testing).

You should ask suppliers to address how they use and handle foreign elements when submitting their tender.

3.2.4 Cross-platform use

Any product or service you procure should allow editable documents to work equally well across platforms (ie, different software, operating systems and

devices). If you create a document in ODF using the proposed solution on one platform, you should be able to read, modify and save it on another - and vice versa. Make sure the supplier gives you the necessary information to test functionality between various platforms.

3.3 Supplier confirmation of ODF support

You should ask potential suppliers to declare one of the following, in relation to whether their product or service supports ODF.

- (a) Yes, our application/solution can use ODF as its default format and complies in full and without any restrictions regarding any of the requirements around ODF.
- (b) By using third party software, through a plugin or add-in, our application is able to read, create and revise documents according to the ODF 1.2 standard. If the end user selects ODF as its default format, he or she will have no restrictions on the functionality of our software based on an incomplete implementation. We are not aware of any loss of information that would occur to documents edited or reviewed by our solution, provided that they conform to the ODF 1.2 specification.
- (c) Another (please ask supplier to provide details).

3.4 Using free open source software based on ODF

The software you procure for working with editable documents must be compliant with ODF regardless of whether it is proprietary software (eg Microsoft Office) or free, open source software (eg LibreOffice). Downloading free software isn't considered procurement by European law if there's no service or support contract and you conform with the terms of use in the licence. Exceeding the licence's limits or procuring additional support and services such as training or development may involve a tender, depending on the value of the contract. In such cases normal procurement rules may apply.

You must still ensure that any free software you use supports ODF. See [Requirements](#) for more information.

4 Base ODF solutions on user needs

How to find the best ODF-compliant document solution for your organisation by understanding your users' needs.

4.1 Plan your user research

To find the best possible document solution for your organisation and plan for a successful transition to using ODF, you first need to understand the needs of your users.

You'll need to identify the:

- kinds of users reading, creating or editing documents in your organisation
- file types handled by each kind of user
- ways each kind of person uses those documents
- [user needs](#) of those people

You'll gather this information using different methods, including internal inventories and user research. You should also get to know your users and office environment, perhaps in a more informal or less structured way.

The process of gathering and analysing this information might also reveal what document-handling practices aren't working and shouldn't be kept, and what new practices you might benefit from.

4.1.1 Identify user types

Most users need software to write a report or a letter, prepare an occasional spreadsheet with some simple calculations or put together a slide pack for a presentation. These users don't need any special functionality and are relatively easy to migrate.

Not every user is the average user, however. Different types of users in your organisation will often work with certain file types. You should try to find a representative group for each type of user to interview so you can better understand their individual needs.

Some examples of specific user types include:

- accountants: all users whose work relates to finance (eg financial analysts, logistical planners, purchase controllers, etc); they often have their own modified document templates and use advanced functionality in spreadsheets
- developers: not only software developers, but also those creating and managing templates; they often use hybrid documents such as text files, database operations and spreadsheets that are linked and where input in one type of file is created by output from another
- scientists: they often use special file types and advanced calculations
- system managers: they often disregard what ordinary users work with and work according to their own methods (eg with scripts)

Interviews with these users generate your [user research](#), which you'll use to determine:

- which file types are used in your organisation
- who uses them and for what tasks

When you conduct user research be sure you can observe what sort of software your interview subjects are using and on what platforms. You can't always rely on users knowing specific terminology or knowing the answers to more technical questions.

4.2 Questions for users: text documents, spreadsheets, presentations, databases

In the interviews, ask your users the following questions:

- To what extent do you currently use word processors, spreadsheets and presentation programmes?
- Do you work with other software that generates or processes documents, or parts of documents? Tell us what it is, if you know.
- On which platforms and operating systems (including mobile) do you currently create documents? And on which do you view documents?
- How straightforward or complicated are your documents?
- Can you share some samples of your documents - some that use special formatting, and some that you use often?
- Do you use templates? If so, do you create them yourself? Can we take a look, or could you send them over?
- Do you use macros? If so, do you create them yourself?
- How frequently do you exchange documents, within your organisation or with people outside? How do you exchange these documents?
- Do you need to see or record changes in different versions of a document? If so, which platforms and operating systems do you use? Do you know of other applications used by people you work with, to edit and review documents?
- What other applications or software do you have experience of using for working with documents - for example things you might use at home?
- Do you publish documents on the web in formats other than HTML? If so, what are they saved as (eg doc, odt, docx, pdf, xls, ppt, csv)? What are the main reasons for publishing the content in these formats?
- Have you experienced migration between different formats before, for example in a product migration from WordPerfect Office to Microsoft Office or Lotus Symphony (or vice versa)? What was your experience?
- What security and confidentiality functionality do you use on your documents?

There may be additional questions for each of the components of an office suite, depending on what document types your users work with most (text documents, spreadsheets and presentations).

Some of these questions will be for advanced users only. You should phrase questions in a way that your users will understand, or demonstrate features and ask if they use them.

4.2.1 Text documents

You should determine which text file types are in use. Make an inventory of the different versions of every file type. For example, a .doc file can be in Microsoft Word 97/2000, in Microsoft WordProcessingML or in another supplier's specific version of either.

Consider asking the following questions or observing your users to find the answers:

- Do you often export documents to HTML?
- Do you use special code, such as VBA, Java or Applescript? Do you use dynamically linked objects (OLE, D-BUS, DDE, ActiveX, OpenDoc etc)?
- Do you use some form of links or shortcuts in documents to refer to other documents in your system environment?
- Do you use templates for icons or graphs?
- Do you use frames, text boxes, active content controls?
- Do you use macros? For what purpose?
- Do you often use the word processor to send email attachments?
- Do you use fill-in forms? For what purpose?
- Do you use automatic spelling correction, dictionaries or modified dictionaries?
- Do you use automatic word completion (sometimes called AutoText)?
- Do you rely on password protection for confidentiality (including opening, changing and printing files)?
- Do you use custom fonts that need to be embedded for people outside of your organisation?
- Do you add accessibility information to the documents you create (eg written descriptions of images)?

4.2.2 Spreadsheets

Determine which spreadsheet file types are used in your organisation. Make a list and also specify the various file types that use the same extension. For example, a file that ends in .xls could be from versions of Microsoft Excel dating back over 25 years, or may have been created by a content management system or mobile app today.

The large variety of different file formats and the problems this causes for users is a significant part of the reason to consolidate document work flows to a single open standard. If you have trouble determining the answer to this question, consult a document format specialist.

Consider asking your users the following questions or observing them working to find the answers:

- Do you use a lot of input from text files to generate data for spreadsheets?
- Do you use formulas for optional parameters?
- Do you typically write complex formulas yourself, or only use existing documents or templates?

- Do you use statistical, financial or mathematical functions?
- Do you use specific features such as Quattro Pro Pivot Tables, OpenOffice.org DataPilot or Microsoft Office PivotTables?
- How large are the datasets that you handle? How is this data managed with respect to versioning, data integrity, confidentiality and security?
- What type of diagrams do you use? For example, bar graphs, pie charts or block diagrams?

4.2.3 Presentation documents

Make a list of the presentation file types currently used in your organisation.

Consider asking your users the following questions or observing them working to find the answers:

- How important are special effects in presentations, such as gradients in 3 colours, double and triple borders, dotted borders?
- Do you use corporate templates for your presentations? How are these maintained? Was there anything missing you've added yourself?
- Do you use voice-overs in presentations?
- Do you use animation in diagrams?
- Do you use mouse-over functionalities?
- Are your presentations sometimes published on the internet? Do you embed copyright or provenance information alongside photos and images?
- Does your organisation aim to be inclusive, ie explicitly require accessibility information to be present?

4.2.4 Databases

You'll need to track down the small, hidden databases inside your organisation. These are often put into place locally to solve an urgent problem. Unfortunately a quick, temporary workaround often ends up being a time-consuming and increasingly permanent one that's likely to cause problems later on.

Developers of these databases (often just advanced users) move on, or forget to inform others that they have a task-specific database running on their machine. In many cases there is no record of the application existing, nor any documentation of its functionality.

If you're making an inventory, don't forget to ask if users collect their own data and, for example, depend on small local databases to save address details, telephone numbers and email addresses. Check the extent to which this category of database plays a role in your IT environment, and consolidate these towards a more permanent solution.

Interaction between a database and text documents, presentation slides and spreadsheets can become complex. Designing and then managing databases securely and sustainably within an organisation requires the in-depth knowledge and professional skills of an IT department.

The basic information needed to connect documents to databases is [specified in the Open Document Format standard version 1.2](#). However, the tools for working with databases is out of scope for this guidance on document formats.

4.3 Select a solution based on user needs

Make a list of functionalities and requirements, taking into account the user needs.

Try to distinguish between how your users currently work, and how your organisation would like them to work. For example, they may still make use of macros even if this goes against best practice in your organisation. In this case you would need to factor in developing alternatives for their use.

4.3.1 Create a shortlist

Create a shortlist of applications you're considering. You can also send out an informal request for proposals, or publish a list of the functionalities your organisation requires and request feedback.

4.3.2 Prioritise functionality

Assess the functionality of each of these applications.

Value	Importance
Must	Describes functionality or requirements that must be included.
Should	Represents high-priority functionality that should be included if it is possible. People in your organisation would experience problems if this functionality didn't work, but could operate if there's a work-around solution.
Could	Describes functionality that's considered desirable but not necessary. To be included if time and resources permit.
Will not	Represents irrelevant functionality, not deemed important enough to justify cost or effort.
Shall not	Represents functionality or characteristics that are explicitly not appropriate and would disqualify certain products or services.

Your organisation needs to cover all the functionality marked as a 'must', and as much as possible of the functionality marked as 'should' and 'could', but it isn't essential to combine all of these in a single solution.

The classification 'shall not' will act as a blacklist. If for example there is a hard requirement for security and confidentiality of data, and a certain solution does not satisfy this requirement, it is not suitable.

4.3.3 Assess

For each of the 3 functionality groups ‘must’, ‘should’ and ‘could’, create a table of potential solutions with 5 columns:

- functionality (concise description of functionality)
- name (the name of the product, service or solution)
- cost (eg estimated total deployment cost of this solution for your organisation)
- platforms (supported platforms)
- notes

For each table, add a row for each of the solutions that fulfills the criteria and fill it out. Make notes of particularly useful, unique features, and important caveats that products have. If you’re uncertain, contact product suppliers or companies that offer support for that product with your question.

4.3.4 Aim for a baseline service

An application or family of applications doesn’t need to address every user need. It’s advisable to establish a ‘baseline service’ for all users, which could mean:

- one productivity solution on each desktop
- one productivity solution on each desktop plus access to a flexible web-based solution that’s accessible on other platforms

When user needs arise that aren’t covered by this baseline service, it can be optimised or added to rather than replaced. Your organisation can ensure adequate support for the baseline service, while allowing power users to work with more advanced solutions.

4.3.5 Identify suitable combinations

Based on the table of potential solutions, identify the top 3 combination of solutions that would be the most suitable for setting your baseline. You’ll need to test these solutions in more detail, to find out if the requirements of your users are indeed met. Don’t forget to check if a product is on the ‘shall not’ blacklist.

Match each solution with the requirements list of the individual users. If you can, ask the actual users who you worked with earlier to help you assess which solutions satisfy their requirements.

4.3.6 Procurement phase

If you’ve selected open source solutions that you’ll be downloading, with no cost attached, you can directly continue with planning your migration and procuring the services necessary to do so.

If you've selected proprietary, paid-for solutions, you'll need to follow a procurement path to access the software. There is separate [guidance on procurement](#) that contains more detail for each of these scenarios.

4.3.7 Users outside your organisation

You'll need to make sure you can continue sharing information with people outside your organisation after you've moved to ODF. Because you're more than likely moving from a vendor-specific format to a format that can be used in many applications, it won't go unnoticed.

Some external users will be working with the vendor-specific formats that most of government has worked with until now. However, there are several modern office productivity tools that work with ODF and are free to access or download.

In many cases users' needs will be better served by providing documents in HTML, which can be viewed on almost every modern device and platform that has a web browser installed. This is often simpler and more cost-effective than installing new software and is more accessible.

Make sure your team is informed about the reasons behind moving to ODF and what its benefits are, as they're likely to get direct questions about it from outside users.

5 Validators and compliance testing

An overview of how validators and testing can ensure that applications dealing with editable documents are ODF-compliant.

5.1 Reasons for compliance

You may need to use either 'validators' or software that compares applications to ensure they're compliant with ODF.

Software applications need consistency so that they can store, share and handle content that behaves in a way that a user expects. The ODF standard was designed to provide that consistency by defining a universal syntax (a shared 'formula') for producing documents.

Products that don't implement the standard correctly are bad for interoperability because they often create a poor user experience when documents are shared across different applications. Other applications then need to correctly deal with an increasing number of violations of the standard syntax. This is expensive and impractical to do now, but may be impossible in the future.

Compliance testing makes sure applications are built to the required standard.

5.2 Use validators to test compliance

It's not possible to run a direct check against a productivity software application to see if it is built to a specific standard. You need to look at the output of applications, for example a text document or a spreadsheet. This can be done with a tool known as a 'validator'.

Validators are pieces of software that perform a series of automated tests to see if a document formally complies with a specification, in this case the ODF standard.

Validators produce a report that shows how a document does not conform to the specification, with a reasonable level of certainty. This helps give you an idea about whether the application (or combination of applications) used to produce the document conforms to the specification.

The ODF standard can be read by people in its text format, but is also produced in a [format that validator software can interpret](#) and that technical experts can use. This is rendered through another standard called [RelaxNG](#), a schema language. Validators can use this to automatically check a document against the standard to determine if it's compliant or not.

Not all checks can be machine-automated in a simple way. Some requirements will need additional dedicated programming to validate.

5.2.1 Which validator to use

No single validator covers all conformance criteria of the ODF specification. There are many validators available that run complementary checks, so we recommend using several in parallel.

The OpenDoc Society has sponsored a free [online validator](#) based on the [ODF Toolkit](#). Other downloadable validators include, for example, [Cyclone3](#) and [Office-o-Tron](#). You can also use one of the many [RelaxNG validators](#).

The OpenDoc Society has more information about [downloading and using validators](#).

5.2.2 Interpret the output of a validator

Validators produce detailed output from the tests they run, sometimes more than normal end users need. Validators aim to be as user-friendly as possible, but reading their results often requires some understanding of the standard.

You shouldn't automatically conclude from errors that are reported by a validator alone that a document does not conform to the specification. Equally, you shouldn't presume that a document conforms to ODF just because it appears to pass a validator test with no errors generated. Validators do not claim to be correct 100% of the time.

Issues highlighted by validators should still be investigated, however. It's important to run a set of representative documents through one or more validators.

Reviewing the output will help you understand the level of standards support across the range of applications that were used to create and manage those documents.

5.3 Use comparison software to test compliance

Comparing how documents are handled in different applications is an alternative to validators for testing application compliance.

Officeshots is a convenient tool for this. It's a collaborative effort of the OpenDoc Society and the government of the Netherlands, and is available as a [free online service](#) and as an [open source tool](#).

If you're considering a particular application, Officeshots can provide an initial but thorough check to see it handles ODF in the way it claims to. It allows you to compare the output and behaviour of a variety of applications, based on documents that are relevant to your organisation rather than on a synthetic benchmark. For instance, do the templates produced by your communications department or design agency look consistent across all the applications they are likely to be used in? How do your outgoing letters look if you view or print them in EuroOffice 2014, different versions of Microsoft Office or Google Docs?

The documents you provide to Officeshots as input, as well as the results from passing these documents to the different applications, are checked in a number of validators automatically. The resulting documents and the validator results can be downloaded, for more thorough evaluation or bug analysis.

Officeshots can run automatically on sets of test documents (ie test suites), or works with other testing methods such as the [ODF autotests](#).

[Source code and documentation](#) are available, as are [sets of test documents](#).

6 Platforms and devices

An overview of the platforms and devices that work with ODF, as well as security and accessibility considerations.

6.1 Operating systems

The Open Document Format (ODF) standard has been implemented on nearly all operating systems for desktops, laptops, mobile phones and tablets, including:

- Android (with many derivatives)
- Apple iOS
- Apple OS X
- Blackberry/QNX
- Chromium OS
- DragonFlyBSD
- eComStation

- Firefox OS
- FreeBSD
- GNU/Linux
- GNU/Hurd
- the Maemo, Mer and Sailfish family of mobile OS
- Microsoft Windows
- Microsoft Windows phone
- NetBSD
- OLPC
- OpenBSD
- OpenIndiana
- Plasma Active
- ReactOS
- Solaris
- Ubuntu touch

6.2 Web-based solutions

Users can also work with the ODF standard through a web platform. This allows ODF to be used on nearly every smartphone, feature phone and tablet currently on the market.

There are 2 classes of solutions based on web technology:

- server-based office applications with a web front-end
- in-browser solutions

Server-based web office applications (such as OX Office, Google Docs, Office 365 and Zoho Office) translate the documents on the server into HTML or another format which can be viewed with a web browser.

In-browser solutions (such as those based on the open source WebODF library) are self-contained and do not require any installation or full internet access. In terms of security and privacy, in-browser solutions are most similar to traditional installed software because:

the actual document is shown and edited directly on the end user device network connectivity isn't required to edit or view documents (only for collaborative working) once the web viewer or editor is downloaded to the browser, there aren't dependencies on software running somewhere else In-browser solutions for viewing and editing ODF can be served alongside the document from environments such as:

- your website
- intranets
- extranets
- content management systems
- document repositories
- webmail

Your organisation can also choose to package an in-browser solution as a native or web app for a phone, or attach an add-in or extension to the browser. Running it from a local copy is much more secure and prevents data loss if you lose your internet connection. The main downside is you need to remember to download updates.

6.3 Security on platforms and devices

Many solutions for different platforms and devices won't offer the necessary security for working with confidential or sensitive documents. Where higher security is required this can be met with:

- locally installed software (including apps) without remote components
- in-browser solutions
- solutions based on a self-hosted web server (ie not accessed through the internet)

When using locally installed software, devices don't need to be connected to offer full document functionality. Local solutions also allow users to automatically apply strong encryption before saving a document (locally or remotely) or synchronising with their cloud provider. Server-based solutions are viable where users always have a reliable network connection and where the server is hosted in a trusted environment.

6.3.1 Connectivity requests

If any software you're evaluating asks for generic permission to connect to the internet you should investigate why it's doing this. Don't assume an application just needs a network connection to display advertisements, unless you've first made sure that your content is secure.

Some software interacts automatically with untrusted remote destinations. For example, many mobile apps for both iOS and Android need to send a document to a remote server in order to convert it from one format to another. This can also pose a security risk.

If you're working with confidential or sensitive documents and require a secure solution you should:

- thoroughly audit the solutions you already use
- inspect the source code of solutions before adopting them (or, preferably, compile it yourself)
- make sure the devices you use offer adequate security
- review any software that's been updated by someone else (e.g. someone maintaining an app store)

See the chapter in this manual on privacy and security (Section 8 - Privacy and security) for more information.

6.4 Accessibility on devices with alternative input

Until relatively recently office applications were used almost exclusively on desktops and laptops, which use larger displays and traditional input devices like a mouse and keyboard. Disabled users have been able to use office applications through accessibility features designed for these systems, often through third-party solutions.

Many smartphones, tablets and other devices use touch-screen technology, which poses new challenges to making documents fully accessible. There are particular challenges for people with impaired visual or motor abilities, in terms of both user interaction and overall user experience.

In-browser solutions can offer increased accessibility, particularly if they're created following the W3C's recommendations for authoring tools. Browser-based solutions are vendor-neutral in terms of the hardware they use - an attractive feature if you don't control the hardware your users use or don't want to be locked into a specific platform.

Some applications are highly optimised for intricate touch usage and are difficult to make accessible. For these it might be worth considering standalone alternatives.

See the chapter on accessibility (Section 7 - Accessibility) for more information.

6.5 Developing platform or device-specific solutions

If you need a solution for a class of devices or a specific platform and none of the existing solutions fit all of your user needs, you could consider developing a custom solution. From a legal perspective, there are no known restrictions in software patents to stop you from implementing the ODF standard on any platform or device or in open source software. You can also adapt open source solutions that are already close to meeting your requirements.

If software or an app exists for another platform there's also the option to port a version to the platform you're using. Existing open source software applications centered around ODF have already been ported to many different platforms. This includes complete open source touch-enabled solutions that can run on tablets and smartphones, such as those based on the [Calligra](#) project.

The technical specifications of ODF are available online for free and there are dozens of software libraries in many programming languages available to work with ODF. OpenDoc Society provides an extensive repository of [example code](#) ready for free reuse.

Most open source projects are eager to work on new features, and many organisations find that the cost for hiring people to develop the features they need is quite low compared to the total cost of ownership of office applications.

If you currently use a non-open source solution on one platform and need to add more features, contact your vendor and let them know what you want. If they are unable to supply these features, consider creating a plugin, extension or add-on to meet your need.

7 Accessibility

How to ensure your ODF document solution is accessible to people with disabilities.

7.1 The accessibility requirement

Many people with disabilities have trouble accessing office documents without assistive technologies or inclusive design. Disabilities that can make working with documents difficult include, but aren't limited to:

- visual impairments
- motor impairments
- cognitive impairments

The severity of these disabilities can vary widely. Visual disabilities, for example, can range from colour vision deficiency (which affects roughly 8% of the male and 0.5% of the female population globally) to total blindness.

You must take appropriate steps to ensure that people with disabilities can access the content of office documents as easily as possible. This can be either through adaptive technologies, such as screenreaders or alternate input devices, or through universal, inclusive design - ie one design directly accessible to those with and without disabilities.

In the UK, to exclude a user on the basis of disability would breach the Equalities Act 2010. See the Service Design Manual for more [information on accessibility](#).

7.2 Make documents more accessible

There are 2 ways to make office documents more accessible:

- appropriate formatting by the author
- automated processes in the applications used to create them

The Canadian Accessible Digital Office Document (ADOD) project has produced [detailed guidance for improving office documents](#) to make them more accessible, including word-processing documents, spreadsheets, presentations and e-books. These guidelines are meant for documents that are:

- intended to be used by people (ie not computer code)
- text-based (ie not simply images, although they may contain images)
- fully printable (ie where dynamic features are limited to automatic page numbering, table of contents, etc and do not include audio, video, or embedded interactivity)
- self-contained (ie without hyperlinks to other documents, unlike web content)

- typical of office-style workflows (reports, letters, memos, budgets, presentations, etc)

There is also [guidance on creating accessible content](#) in the UK Government Service Design Manual.

7.3 Product-specific support

Some productivity tools have better support than others for the accessibility features included in the Open Document Format (ODF) specification.

Tools with high quality support include:

- Microsoft UI Automation with Microsoft Office
- the combination of IAccessible2 API with Apache OpenOffice (as of version 4.1) and LibreOffice (as of version 4.3)
- Microsoft Office Online (part of Office 365), which provides WAI-ARIA support for assistive technologies (WAI-ARIA stands for Web Accessibility Initiative-Accessible Rich Internet Applications)
- Office Online where you use an assistive technology such as a screen reader or speech recognition software, if the assistive technology supports WAI-ARIA

Previously included as part of Microsoft Office, speech recognition is now available as part of the Windows operating system. If you use Microsoft Windows XP and need speech recognition, you need to use an older version of Microsoft Office with the ODF converter plugin.

AccessODF is a useful extension for Writer (the word processing application in LibreOffice and Apache OpenOffice) that helps improve the accessibility of text documents. Recent versions of Microsoft Word include an integrated checking function.

Product-specific information on accessibility is available for:

- [Apache OpenOffice](#)
- [Google Apps](#)
- [LibreOffice](#)
- [Microsoft Office 2013](#)
- [Microsoft Office 365](#)

7.4 ODF support of alternative media

ODF supports braille, large-print, and audio alternatives to print and e-documents. Audio is particularly important, as it provides access to documents for many people with disabilities such as visual impairment or blindness and learning disabilities.

When implemented correctly, these and alternative input methods will help users with disabilities to read, create, and edit documents - with full access to all of

the meaning and intent. A blind person, for example, should be able to work with a document someone else created by getting a text description of the images used. A person should if needed be able to fill out a form without using hands. Someone with poor vision should be able to read presentation materials easily.

Automatic open source converters are available for easy conversion from ODF to braille, and from ODF to the DAISY audio book standard.

7.5 Build solutions that implement accessibility

While ODF offers accessibility features, whether they're available and how well they'll function will depend on how successfully the standard is implemented in applications and assistive technologies.

Many blind users depend on third party assistive technologies, many of which are hardwired to a limited set of established products and platforms. These products should continue to support user workflow seamlessly if they support ODF 1.2 well.

If your organisation selects other products, make sure adequate assistive technologies are available as well - or allow exceptions.

It's relatively straightforward to support the main accessibility features such as text labels and object descriptions coming from an application. Find out which assistive technology requirements your users have, and talk to vendors. If you don't have the technical capabilities in your organisation to do so, pool together with others or hire experts. Build in enough time and think ahead: this will allow for the most cost-effective solutions. Vendors will often be receptive to reasonable bids for making their existing solutions more accessible. Accessibility is something both vendors and open source communities want, but often lack the budget for.

7.6 ODF accessibility guidelines for developers

The ODF technical subcommittee on office accessibility has ensured that versions 1.1 and 1.2 (the current version) of the ODF standard support encoding to make documents accessible through assistive technologies. There are detailed [accessibility guidelines](#) for version 1.1. Written for developers, the guidelines include detailed information on approaches to accessibility for different categories of disability. They're also a useful resource for anyone interested in accessibility issues for productivity software.

8 Privacy and security

How to ensure your organisation's privacy and security when using ODF document solutions.

8.1 Minimise risks to the security of documents

Working with and sharing office documents poses potential risks to the privacy and security of both documents and user systems and accounts.

Threats to document security include:

- unauthorised interception of an emailed document
- unauthorised access to a document on a computer, information media or network
- compromises to content integrity

The main threat to user systems and accounts posed by documents involves macros, which can be used by attackers to take over a user's computer system and should be avoided (see section 9 - Avoid macros in documents).

Open Document Format (ODF) offers 3 ways to minimise risks:

- document encryption
- protection locks - i.e. passwords to restrict certain activities
- digital signatures that verify a message is authentic - i.e. sent by the named author and not altered in transit (often used in monetary transactions)

8.2 Document encryption

Document encryption completely locks a document from unauthorised users. Users employ the recipient's public key when composing a message, which they'll then decrypt using their private key.

When an ODF document is password protected the file structure of the bundle remains the same, but contents (of XML files in package) are encrypted using the following algorithm:

1. The user-created alphanumeric password is converted into a digest, a single string of digits generated by applying a formula called a one-way hash function to the password.
2. The local time on the computer is used as seed (ie initialisation) for the production of a random 8-byte initialisation vector (the number used along with the secret key to encrypt data).
3. This process also produces a 16-byte salt - an additional piece of random data used with the hashed password to create a unique key for each file.
4. This unique key is then used together with the initialisation vector to encrypt the file in cipher-feedback (CFB) mode (a particular algorithm used to enable encryption). The initialisation vector and salt are both stored in the manifest file.

ODF versions before 1.2 used SHA-1 (a secure hash algorithm) and Blowfish (a symmetric-key block cipher). This combination fails to meet several government security standards (eg FIPS-140).

ODF 1.2 allows other algorithms to be used. For example, Apache OpenOffice (as of version 3.4) and LibreOffice (as of version 3.5) can also use SHA-256 and AES (Advanced Encryption Standard) for encryption. As of early 2015, this combination is considered reasonably secure, as unless someone has the password the document can't be opened. If you lose or forget the password it's extremely difficult, if not impossible to recover the document's contents.

8.3 Protection locks

You can also lock specific parts of text with a password to prevent activities such as editing. Once you lock a section, any user attempting to access it will be asked for the password. This is called a protection lock. Protection locks offer only a low level of security because someone with know-how could edit the XML files to bypass the password and access the document. They can, however, prevent users from accidentally altering the text.

8.4 Digital signatures

Digital signatures in ODF verify:

- the authorship of a document
- the integrity of a document - ie that it hasn't been modified without authorisation

ODF supports digital signatures using public key technology, so no private passwords need to be exchanged. The author of an ODF document uses their private key to sign it. The recipient uses the author's public key to check its authorship and integrity.

Digital signatures are reliable, especially when the same software product is used to verify a document as the one used to sign it. Digital signatures in the ODF 1.2 specification are different from the non-standard signatures introduced as extensions by implementations of earlier ODF specifications.

Signing by ODF implementations is performed on the saved document. Any number of signatures can be affixed, by different parties at different times, providing a form of counter-signing.

The standard implementation of digital signatures only confirms authorship and integrity. XAdES (XML Advanced Electronic Signatures) extensions used in ODF digital signatures carry further information about the role of the signer and the purpose of the signature. While few implementations support specification and presentation of the XAdES data, such signatures remain verifiable and XAdES should be used wherever support is available.

Again, consult the guidelines and policies for official documents for:

- the use of public-key certificates
- required signing techniques (such as the use of time-stamp systems)
- the treatment of digitally-signed materials

8.5 Save with password

Some ODF-compliant applications allow documents to be saved with an associated password, which must then be entered to view or access the document. This ‘save with password’ feature uses [digital encryption](#) as defined in the ODF specification. This offers limited security that’s useful for preventing accidental disclosure of personal documents to others. ‘Save with password’ does not protect related documents, auto-saved files and related files. To protect these files, consider encrypting the hard drive or the section of it containing them.

Protecting individual documents with a password has several weaknesses:

- if the document creator doesn’t choose a good password it could be guessed or determined using software
- sharing a password-protected document involves separately sharing the password with the intended viewer/editor; if this isn’t done securely the password could be intercepted.
- the password could be inadvertently disclosed in another way by the recipient
- the document’s plaintext could be disclosed by the recipient

Passwords should:

- not be reused for other purposes - even repeated for the encryption of other documents
- be strong
- be created using password-generation utilities if possible

Encryption for ODF documents is inherently insecure because someone with sufficient knowledge of encryption might be able to determine the encryption key (different than but derived from the password) just through access to the encrypted document. With this key and enough determination, an expert in encryption methods could decrypt the document.

Because of these limitations of ODF encryption, sensitive documents (even those graded ‘official’) that are shared or otherwise might be available to third parties should be secured using other measures.

8.6 Choose a good password

It’s important to choose a good encryption password. It must be both:

- complex and not obvious for others to guess
- simple enough to remember Don’t use words from any language. If you use a word found in a dictionary, your security will be susceptible to what’s known as a ‘dictionary attack’. Include numbers, punctuation and other symbols throughout your password to improve its security. Longer passwords are more complex and better able to withstand malicious software that systematically checks all possible keys or passwords until the correct one is found.

8.7 Work safely in the cloud

Cloud computing involves storing and accessing data and programs over the internet instead of on your computer's hard drive. If your documents are stored on the server unencrypted or (as is the case with solutions like Google Docs) if your keystrokes leave your browser, their contents should be considered insecure.

The cloud can, however, offer security if you can control where you store the content and can encrypt everything before it leaves the browser. The RemoteStorage open protocol allows users to choose where to house their data, whether with a trusted provider or their own storage server.

9 Avoid macros in documents

Why macros shouldn't be used in documents and what alternatives ODF can offer.

9.1 What macros are

Macros are small pieces of software included within a document to automate certain repetitive tasks, such as alphabetically sorting items in a bulleted list.

Macros offer features that make them both convenient and powerful, but they also have problems with interoperability and security. Macros that work on one version of software or platform won't necessarily function on another. They can also allow viruses to infect your computer and distribute themselves using an application's own programming language. Even when a file comes from a known and trusted person or organisation it may already be infected without their knowledge.

For these reasons the UK government forbids the use of macros in publicly shared documents and advises against their use in general.

9.2 Alternatives to macros

Web-based document solutions provide the same interactive and automatic functions as macros do, but without the interoperability and security issues.

Office applications can nearly always offer the same functions as macros through extensions, add-ons and plugins. If the right one doesn't exist, you can get one built.

Before using extensions, add-ons or plugins you should:

- test them for interoperability and usability to make sure there are no unintended side effects when you share documents across platforms
- get them vetted by IT security professionals
- restrict levels of access for certain privileges to developers or system administrators (give users content-level access only)

9.3 Assess your organisation's macros

Before you look for an alternative to a specific macro you should first determine if it's meeting a user need.

First run an automated scan for macros on system drives. You may discover that many macros are undocumented, quick workarounds created by a user to complete a one-off task. There may also be more than one macro created by different users but for the same purpose. Recreating most of these macros as a plugin would not be worth the cost.

You should also ask your users what functions they would be missing if document macros are disabled. Ask them to list those macros they actually use. One German government agency counted 9,000 macros on disk but found that users only needed 9.

9.4 Plan your move away from macros

Moving your organisation to open file formats gives you a good opportunity to assess and improve its processes related to documents.

Start to determine how widespread macros are in your organisation and what functions they provide. If macros are widely used don't try to stop using them all at the same time. If something goes wrong during this kind of 'big bang' approach, your users might be confused about what is causing the issue.

Once you have an overview of which macro functions need to be recreated in plugins or applications, begin planning the order in which you'll turn off document embedded macros. You should also work out a budget for doing so and involve IT security staff to make sure you understand related security issues.

9.5 Deal with macros post-migration

To prevent your users from unintentionally using macros, disable them in the settings for all applications. You may also be able to scan incoming documents and remove macros before they reach the user.

If your staff can't avoid using macro-embedded documents at first, provide a standalone (preferably offline) machine where users can take macro-embedded documents on a thumb drive and process them in isolation from other systems.

10 Integrate ODF with enterprise tools

How ODF fits into the way your organisation processes documents.

10.1 Enterprise tools

Enterprise tools are software applications that perform specific processes that an organisation needs, rather than its individual users, such as record and document

management. Office documents will often be processed in some way by enterprise tools, which can involve:

- pre-processing (eg creating a template letter using information from a case management system)
- post-processing (adding an electronic signature or annotating the document)
- other kinds of manipulation or modification

Enterprise tools are often embedded in an organisation's working practice. They'll need to be carefully integrated with your document solutions to ensure they work with Open Document Format (ODF).

10.2 Choose a platform

You'll need to decide what platform you'll use to integrate enterprise tools with ODF. Your choice should depend on your users' needs.

Web-based solutions have advantages over office applications. They

- don't require installation or updates on your user devices
- are less complex
- are available for use on devices other than just desktops
- are less prone to user error

If you need convenient access to rich text editing, making charts or adding corporate branding then web-based solutions might not be appropriate. Office applications can offer these, but there are also dedicated software components that allow you to integrate office viewing and editing capabilities into your own software. For example:

- libraries such as the open source [Calligra](#) Engine or [Aspose](#) natively work with ODF and allow easy integration
- into desktop, server and server-side web applications
- the open source [WebODF](#) framework is built around ODF viewing and editing within client-side web applications

10.3 Automate tasks and processes

If you're using office applications in combination with macros to assemble documents and carry out tasks, you can probably do the same thing more easily and cheaply with ODF. You should move avoid using macros when possible to avoid issues with security and interoperability. ODF allows you to assemble documents and automate:

- different stages in a process
- repetitive tasks

- reports or presentations from business data

Many software libraries can process documents from different sources, allowing you to assemble and manipulate the content in different ways. Most are easy to use, offer advanced functions and require little knowledge of ODF.

10.4 Records and documents management

Keep structured information intact and accessible in an automated way, close to the original source of the data. There should be one authoritative source where all information that belongs together is maintained.

For instance, if you have a collection of documents that have been created by mail-merge from a database, you should save the database as your record, not the individual letters in their document format. Using one or more office applications to maintain separate, stand-alone documents makes it much more difficult to manage that information or get a clear overview of it.

10.5 Use metadata to record a document's history

Once your content has been moved into ODF, you should keep information well-structured and complete. For quality assurance you need to trace where your information is modified, by whom or what, and how. You can do this by using metadata, which is data about other data - in this case describing the information in your document. Metadata features in ODF allow you to keep track of what happens inside your documents, from the moment they're generated until they're archived.

When attaching metadata you should:

- specify which document template was used to create a series of documents, so you can make updates easily when the template is replaced
- make it clear what part of a document is modified by automated software and what version of the software you're using by setting the meta:generator tag or other relevant meta-tags
- enable version control (change tracking) by default where a person will modify a document generated by an application; you can see the history of the document, scan for errors and other integrity issues and trace the impact of mistakes throughout the entire application chain
- add a digital signature to a document if the integrity of information contained in it is important
- place scripting or placeholder instructions in a logical place, preferably inside user variables or script tags

10.6 Create new ODF templates

When creating new ODF templates make sure that:

- they are clean and technically sound
- they have been verified to work in the most common applications
- fonts and sizes are included in the template

Add the appropriate accessibility information for anyone who will be working with your document, including people who need assistive technology.

If you have to convert legacy templates you should:

- validate every step in the process
- make sure the manifest is complete
- make sure all reference styles are defined

See the section about validators and compliance testing (section 5 - Validators and compliance testing).

10.7 Replace legacy applications incompatible with ODF

You should consider replacing legacy applications that can't work with ODF. Most of the tools needed to work with ODF are low-cost (or free) compared to traditional solutions and office applications. As an interim step, create documents using XML, JSON or structured text output using one of the tools available in the software library.

If no compatible outputs are available from your product, you may need to work with the best possible option your vendor provides and then create the output in ODF as you need it. Many office applications offer a non-interactive batch mode that can be used for conversions. Alternatively there are external tools like OfficeConvert for Microsoft Office, which was produced for the ODF validation service [Officeshots](#).

Converting documents from legacy formats can be inefficient but manageable unless your output is unpredictable. Validate your output and clean up the results to prevent processing errors in future.

10.8 Don't use direct formatting

Never manually change the font type, font colour, background or letter spacing in template or document generation software. You should always use styles and headings, otherwise there will be information loss for people using assistive technology, search engines and PDF generators, among other tools.

10.9 Variable data printing

Variable data printing involves printing very large numbers of documents (ie hundreds of thousands or even millions) generated from, for example, large mail merges. If your organisation does variable data printing, the content of your ODF compatible office documents will need to be converted to [PPML](#)

(Personalised Print Markup Language). PPML is a dedicated standard for high performance printing of large volumes of nearly identical documents. Like ODF it's an XML language, which allows standard XML technologies such as XSLT to automatically convert parts of documents created in ODF to PPML-ready assets.

11 Extensions, plugins and custom solutions

How to ensure that software offering added functionality to document solutions meets the ODF standard.

11.1 Additional software options

If your office productivity tools don't offer all the functions you need, additional software - extensions, plugins or add-ons (depending on the application you're using) - can help.

If you're using one of these to help create or manipulate content you should be aware of any impact this has on the output document's conformity to the Open Document Format (ODF) standard.

If you're responsible for developing an extension, plugin or add-on, consider using one of the [existing libraries that work with ODF](#).

11.2 Conforming documents

ODF 1.2 compliant documents don't all conform to the standard equally. There are 2 classes for conformance:

- conforming documents
- extended conforming documents

You should produce conforming documents where possible. Using extended conforming documents may cause interoperability problems and adds complexity to documents. For instance you have to make sure that your applications can flag the document and add information to the document metadata to point to relevant information.

In most cases, extensions, plugins and add-ons to office applications can insert external data in a manner that allows a document to conform to the standard without resorting to extended conformance.

11.3 Image or graphic plugins

Plugins that insert visual information should always use the universal mechanism for claiming a draw area within the document to embed multiple alternative renderings of objects. This ensures that the content can be viewed by any

conformant application and provides an editable document that's portable across applications.

You should provide:

- a vector version (using the web standard SVG) for sharp images
- a bitmap version (using the web standard PNG) as baseline
- your custom file format (encoded as base64)

Ensure that content is accessible by adding proper accessibility information (eg a title and summary description) to new objects. If the plugin has a user interface, make sure that is accessible as well.

If you use a commercial plugin, ask your vendor to ensure it's accessible. If you'd like to use an open source plugin and it lacks full accessibility features, contribute any that you develop back to the project - or commission the developers to add this relatively simple functionality.

11.4 Validate output

If you use an add-on, plugin or extension to create or manipulate content, be aware of any impact it might have on the conformance of your document output. Use an ODF validator (see section 5 - Validators and compliance testing) to check the output.

12 Collaborate on documents

An overview of the different options for collaborating on ODF documents.

12.1 Applications that support collaboration

Document solutions that implement Open Document Format (ODF) can allow different people to work together on the same document. These collaboration features can appear in:

- on-site office document applications such as those found in productivity suites (eg LibreOffice)
- open source embeddable software components
- applications run from a server (held either on-site or in the Cloud) and accessed with a web browser or desktop application

How your users share and collaborate will help determine which of these is best for your organisation.

Desktop productivity suites are probably the most familiar of these to users, but they won't necessarily be the best match for their needs. They're often complex

and contain features that can be confusing, while their flexibility and portability across platforms is limited.

One of the many open source and commercially embeddable software components or out-of-the-box solutions should be able to offer users the tools they need to work effectively. You're likely to find the right one for your users' needs, whether they need to:

- use a template to compose a document that team members can easily review
- create or manage text documents contributed to by different authors
- exchange information directly with your enterprise resource planning (ERP) system or customer relationship management (CRM) database

12.1.1 Web applications for collaboration

If you need to collaborate with other people on shorter text documents, a solution based on web technology is worth considering. Web applications are more:

- flexible - you can use them with people inside and outside the organisation
- convenient - every modern networked device has a browser so you don't need any installation or additional training
- cost-effective - you don't need to pay for more applications

A web browser can be used directly for document collaboration, but there are also specialised web-based collaboration tools. If you're writing longer generic text with multiple authors, such as a book or brochure, consider using a web-based writers' platform such as [BookType](#). It uses open source software web technology that allows authors, editors and other stakeholders to work and communicate simultaneously through the browser. The result can be exported to ODF, HTML and PDF, as well as various electronic book formats such as [EPUB](#).

For collaboration involving a larger number of participants, an open source web commenting tool such as [Co-ment](#) can gather input efficiently and transparently, in a way that office applications can't.

12.2 Collaborate offline

Offline review allows users to make comments or edits on a document regardless of whether they have a web connection, and then later share the document with other collaborators. ODF standard offers 2 basic offline review options:

- adding annotations (remarks or notes)
- using change tracking

Both annotations and change tracking can be removed automatically by web publishing software. You shouldn't add remarks inside the body of the text, as they might be left in the text and published.

12.2.1 Annotations

Annotations are textual notes attached to the content of a document. They're a reliable and stable way to collaborate and work consistently across almost all applications. The content of annotations is also available through assistive technology, such as a Text-To-Speech reader.

Open source tools like [Docear](#) and [I, Librarian](#) can sort documents and annotations into categories and let you view multiple annotations of multiple documents, in multiple categories, all at once.

12.2.2 Change tracking

Change tracking lets you suggest wording within a text that the document's author may then accept or reject. Change tracking has benefits, but there are also risks you should be aware of. Not all applications implement it as a feature. See the separate chapter on change tracking (section 13 - Change tracking in ODF).

12.2.3 Pre-publication tools

Documents that have been reviewed offline carry more information than you may realise. If you publish documents on your website, you might want to remove change tracking and annotations using an open source software library such as [lpOD](#).

Or you could use an ODF-aware commercial application like:

- [MetaClean](#)
- [Metadata Assistant](#)
- [Datadistiller](#)

Some of these vendors also provide tools to actively scan and remove the change tracking information and annotations, for example in outgoing email.

12.3 Collaborate in real time

You might want multiple people to be able to edit a document at the same time, as this can save time and be more efficient. This is often called 'collaborative real-time editing'.

Real-time collaboration solutions avoid many of the issues that users experience with offline reviewing. They offer an easy way to share, and some allow access to documents to be controlled.

Applications that allow real-time collaboration can be set up in different ways:

- peer-to-peer (directly between users)
- server-based, hosted locally within your organisation
- service-based, hosted by a third party

12.3.1 Peer-to-peer (p2p)

Some ODF applications can set up collaboration directly between the devices of end users, using the same software. This is done using instant messaging or over a direct IP link. Combined with strong end-to-end cryptography p2p offers privacy and confidentiality for multi-person collaboration - only the authors and reviewers will have access to the document.

Collaborative ODF-compliant editors capable of p2p include:

- [WebODF](#) (p2p version not yet released)
- [AbiWord](#)
- [LibreOffice](#) (p2p version not yet released)

12.3.2 Server-based with local hosting

Some ODF applications can be hosted locally - ie within an organisation and connected through a local network. These include:

- web applications such as:
 - OX Apps
 - OwnCloud
 - EtherCalc
 - Bread
 - kotype
- desktop clients such as:
 - AbiWord
 - (experimental) LibreOffice

The web applications are a particularly convenient solution for both internal and external use. They don't need to be installed or updated on any device and can be used with a wide range of client devices.

12.3.3 Service-based, hosted by a third party

There are some well-known ODF-aware solutions available 'as-a-remote-service' only, such as:

- [Google Docs](#)
- [Microsoft Office Web apps](#)
- [Zoho Office](#)

All of the applications that can be hosted within the organisation are also available as a (supported) hosted application, or can be used with any typical cloud provider.

If you use third party web tools hosted outside your organisation, you need to consider security and privacy, both for the organisation and for external users. In most cases every keystroke of every participant is sent outside of your organisation, and may well be logged elsewhere.

12.4 Bespoke solutions

If none of the available solutions for collaboration fulfill your needs, you could:

- build on their capabilities by adding additional features
- engineer a tailored solution

Which option is most suitable for your situation depends on the needs of your users and the volume of their work.

A tailored solution might, for instance, combine document creation and editing tasks with web applications already in use in your organisation.

If you commission a bespoke solution, make sure you get the work with an open source licence to avoid future lock-in.

Try to use generic, existing open source components wherever possible, and contribute custom developed features back to the projects you've sourced them from.

For example, the open source WebODF adds ODF viewing and editing capabilities to:

- webmail clients
- content management systems (CMS) and cloud storage
- records editing in Computer Assisted Qualitative Data Analysis Software

You'll be contributing back to so-called 'upstream' open source projects, which makes the work your organisation funds reusable for others. You'll also help keep the development of these components synchronised with the software you develop.

The chapter on support and training (section 17 - Support and training) includes useful related guidance.

13 Change tracking in ODF

An overview of the change tracking options in ODF: their benefits, risks and alternatives.

13.1 Options that support change tracking

Change tracking lets reviewers suggest new wording, which the document's author may accept or reject. It's an optional feature of the Open Document Format (ODF) standard, which is supported by many open source editors including:

- [Apache OpenOffice](#)
- [EuroOffice](#)

- [LibreOffice](#)
- [NeoOffice](#)

It's a relatively complex feature to implement.

Only some current versions of Microsoft Office can record change-tracking information when saving a document to ODF. If a user adds tracked changes in an unsupported format, these may be accepted automatically and not tracked when the document is converted to ODF.

Older versions of Microsoft Office that used the open source ODF Converter Add-in did have change tracking capabilities, so if your users are on an older version of Microsoft Office and need the feature, the source code of the add-on is available.

13.2 Risks and benefits

Change tracking has both benefits and potential risks.

The benefits are that:

- a reviewer's suggested changes can be approved or rejected quickly
- an editor can easily switch between displaying the text with proposed revisions made and the original text with the changes suggested

The risks are that:

- change tracking might be applied inconsistently, as users can easily forget to turn on change tracking, or may have turned it off and on again
- changes might not all be correctly captured, especially if people are using different applications, old versions of software or mobile apps
- a document might be infected by a virus or other malware
- the change tracking log might be incomplete: other, unlogged changes might have been made

13.3 Automatically compare documents

There are also ways to compare different versions of ODF documents so you can see what changes were made. The different options for this include:

- through a feature available in some office applications
- standalone applications or services, such as tools that highlight all the differences between 2 versions of a document for manual inspection (eg [DeltaXML ODT Compare](#))
- other standalone solutions (eg a plugin that can be integrated into an office suite)

13.4 Automated multiway merging

If your organisation sends longer documents to many reviewers, manually comparing them may not be efficient because the editor will receive too many modified copies containing changes or textual suggestions.

Automatic merging software can be a solution in this case by allowing you to:

- accept or reject changes from multiple reviewers while working on just one document
- group small adjacent changes together for easier comparison
- optionally ignore markup changes when they represent disallowed formatting

Automated multiway merging can be particularly useful when you're:

- working on large documents with multiple authors and need strict control
- resolving conflicts that might occur when multiple users check in the same edited document
- rejoining independently authored versions of some original text

13.5 Alternatives

Alternatives to change tracking include annotations, real-time collaborative editing and web-based collaboration tools. See the chapter on document collaboration (section 12 - Collaborate on documents).

14 Embed fonts in ODF documents

How to embed fonts in ODF documents, and guidance on choosing fonts.

14.1 Open licence fonts

To help ensure documents look the same to all viewers no matter what device or platform they're using, always embed any specific fonts used to compose them.

Open Document Format (ODF) lets you embed fonts inside any document. To make the most of this feature, you should use fonts with an open licence as the default fonts in your applications.

The benefits of 'open fonts' are that they help you:

- reduce costs (some fonts can be very expensive to install and license for each user)
- avoid documents being inaccessible or 'broken' to some users because of licence restrictions
- avoid legal compliance problems

- have more control over how your documents look to all users
- keep more accessible archives

You can get open fonts from:

- most Linux/UNIX distributions such as [Debian](#), [Fedora](#), [OpenSuse](#) and [PC-BSD](#)
- many open source office applications
- non-commercial open font resources such as [Open Font Library](#) project and [CTAN](#)
- [Google Fonts](#)

Whether you use open or restricted fonts, you need to set a clear policy on which fonts you use and make it available to users.

14.2 Freeware and shareware fonts

Freeware and shareware fonts are also available online. These are usually distributed for free but their creator expects a payment when you start using them for business purposes.

They may or may not be safe to embed and modification (eg to accommodate adding a new currency glyph or diacritics) probably won't be allowed.

If you want to use a freeware or shareware font because you can't find a similar open font, you should first check its licensing conditions. These can usually be found online.

14.3 Microsoft and Windows fonts

Some font files are restricted by specific commercial licences. For example:

- the default typeface used in the recent versions of Microsoft Office - the sans-serif typeface Calibri
- fonts distributed alongside the latest versions of Windows and Microsoft Office - such as Cambria, Candara, Consolas, Constantia and Corbel

These ClearType Font Collection fonts pose significant problems to users on other platforms, including most smartphones and tablets.

So if you're using these fonts you should embed them (embedding is allowed according to the licensing information Microsoft has published on its site).

You can't embed the Microsoft fonts Arial, Arial Narrow, Times New Roman, or Courier New.

14.3.1 Check if you can embed Microsoft and Windows fonts

For Windows users there is a free tool called [font properties editor](#) that allows users to check if a particular font can be embedded. This check is based on the information stored within the fonts themselves.

For fonts produced by Microsoft itself, you can also consult the website of [Microsoft Typography](#) for licensing information and how to check compliance.

14.3.2 Alternative open fonts

Many fonts can be used for embedding without any issue:

- the open font [Carlito](#) is a sans-serif font that can be used as a drop-in replacement (ie font-metric compatible) for Calibri and can also be used online
- the [Caladea](#) open font is metrically compatible with Cambria
- the so-called [Liberation fonts](#) can be used to replace the previous generation of fonts used by Microsoft: Arial, Arial Narrow, Times New Roman, and Courier New

14.4 Add a font manually

ODF lets you embed fonts ‘as is’, meaning that technically the entire font file is put inside the document without making any changes to it. This allows a user to take an embedded font and install it on their system. ODF places no restriction on the user to embed or use any font, and leaves it entirely up to the user to deal with this capability responsibly and legally.

You can embed fonts manually by following a few simple steps or by using Fontos, a handy open source tool that allows you to see what fonts your documents are using and to drag and drop them using a web browser.

15 Corporate styles and templates

Best practice for working with styles and templates in ODF.

15.1 Styles and templates

Open Document Format (ODF) offers advanced style and template functions across office document applications. You should be aware of general good practice and specific considerations for working with and managing styles and templates in ODF.

15.1.1 Styles

A style is a bundle of formatting properties that can be applied to a specific document element or any part of it. It can specify typeface, font size and characteristics such as:

- where elements are placed and what the context should look like, including margins, padding, borders, backgrounds
- page breaks before and after an instance of a certain style
- specific language for the element involved
- whether a certain type of numbering needs to be applied
- automatically generating a table of contents from the headings in the document

Creating and applying styles to documents instead of directly formatting them enables you to:

- maintain a consistent corporate identity in documents
- maintain a document's structure
- create documents that are accessible to people using assistive technology
- automate the process of converting documents to other types of content eg DAISY audio content
- help people focus on creating good content rather than managing style and formatting

15.1.2 Templates

Templates are blank documents pre-configured with text and style elements that are relevant and common to particular document needs. Your organisation will have a set of styles used across templates and documents.

15.1.3 Styles and templates in ODF

ODF applications typically come with a set of predefined styles, such as 'Heading 1', 'Title' or 'Footnote'.

A paragraph style can be set to automatically have another style follow it. 'Heading 1', for instance, is usually followed by a style called 'Content Body', 'Default' or 'Normal'. Companies can add their own styles to a template.

The OASIS specification of the ODF file format has text styles similar to 'variable name' and 'variable default value', which allow values to be automatically extracted from the document.

ODF template documents have the following file extensions:

- .ott (texts)
- .ots (spreadsheets)
- .otp (presentations)

- .otf (formulas)
- .otg (graphics)
- .otc (charts)
- .oti (images)
- .oth (HTML output)

15.2 Create templates and styles

When creating templates and styles:

- begin with a blank template and keep interoperability in mind
- prefer features that are mandatory for conformant applications over optional features in the standard
- avoid constructs that don't add much value, such as frames in frames
- include proper fallbacks for images.
- ensure templates are suitable for users of assistive technology
- use a validator to make sure template documents fully conform to the ODF specification
- test templates and documents in the important applications people inside and outside your organisation - work with (the [Officeshots.org](https://officeshots.org) tool can be used for this)
- embed any fonts used that might not be available on every platform
- train users how to work with templates and styles

15.3 Keep templates up to date

Organisations typically use many templates, most of which will be derived from a smaller set of root templates. When these root templates need to change all the templates that were derived from them will also need to be changed.

You may need to update installed templates to allow for:

- new features
- design changes
- interoperability issues

When multiple versions of the same template exist it can be unclear which documents use which version. Adopting a proper naming and versioning scheme for your templates can help avoid this confusion. This can, for instance, allow your organisation to later re-apply a new version of a template automatically when the corporate style is updated.

When you've used a template to create a new document, an application can [link to the template](#) (or a specific version of that template) inside the document metadata. You may also want to [insert the name of the template](#) in the visible text of a document (eg in a footer) to help identify the template used where it might not be otherwise obvious (ie in a print flow based on a number of different templates that sometimes vary).

Organisations using many templates in a complex hierarchy should consider managing them with a dedicated standalone tool. An online ODF template generator such as [DocGen](#), for example, can allow different departments to create custom ODF templates using a pre-approved set of variants. Log custom templates so updates can be made to the root templates they were based on. New custom templates can then be automatically generated and distributed. This solution directly interfaces with office applications but can also be integrated with document lifecycle management tools and mid-office applications. Browser-based tools are also available.

15.4 Dynamic content in templates

ODF applications let you insert dynamic content in a template using

- variables (the most common option and a fairly reliable one)
- form functionality
- preprocessor or plugin

Embedded macros aren't good practice.

15.4.1 Variables

You can create templates to automatically insert or hide specific information depending on the context. Contextual variables might include:

- a page number
- the number of pages in a document
- the current date
- user-defined variables from the document's metadata or other source the application is aware of

Variables can be displayed in a document using so-called variable fields, but can also be used as conditionals (ie if/then determinants).

15.4.2 Form functionality

Sometimes a template requires input from users to generate documents. ODF includes a form functionality that permits this. Based on a standard called [XForms](#), form support in ODF allows high levels of interaction, is flexible and safe, and can be charged on a per template basis.

Your organisation could also consider:

- using a web application (browser-based tools are better than document formats for collecting information)
- building a custom application with an office library like [Calligra](#) or [Aspose](#), to provide a streamlined user experience that avoids the use of an office application altogether
- integrating with the office application using a plugin or extension

16 ODF spreadsheets and formulas

Important considerations when working with spreadsheets and formulas in ODF.

16.1 OpenFormula for data manipulation

Spreadsheets perform many functions for users, from repetitive calculation tasks on ranges of data to storing and sorting ranges of text information. They offer a simple data entry and lightweight programming environment that allows users who lack programming knowledge to perform simple data manipulation tasks.

OpenDocument Format 1.2 includes OpenFormula, a standardised set of formulas that should be present in every compliant application. Formulas are small software routines in spreadsheets that allow users to perform tasks. They allow a combination of:

- fixed parameters, text strings and logic operators
- data (ranges) from elsewhere within the spreadsheet or its metadata
- (selection of data from) another source such as a database
- the result of the computation of other formulas

Because OpenFormula operates across all Open Document Format (ODF) applications, when you cut and paste part of a spreadsheet into a document, it will retain the actual logic behind the cell values rather than just copy their context as text or image.

Using OpenFormula also prevents different formulas with identical names from being confused when users operate between different proprietary spreadsheets. No special converters or compatibility modes are needed. Versions of the standard before ODF 1.2 may still have issues when interacting with Microsoft Excel or other spreadsheets, so be sure to update to the most current version of the standard.

16.2 Levels of OpenFormula implementation

OpenFormula can be implemented at 4 different levels, each offering progressively more capability:

- minimum: providing the minimal capabilities (including functions, types, and their meanings) that are very widely implemented with spreadsheet applications, even in resource-constrained environments (this includes about 100 functions)
- minimum desktop: providing the minimal capabilities expected from typical desktop spreadsheets
- basic: providing all the capabilities necessary for typical desktop spreadsheet use (adds a distinguished logical type and support for complex numbers)

- full: providing some additional, less-commonly used features intended for advanced users

The minimum and minimum desktop levels provide all the functionality the vast majority of users will require. They're widely implemented by desktop spreadsheet applications.

When implemented at the basic level, an application supports all the functions supported by equivalents like Microsoft Excel.

Apart from the extended range of formulas, basic and full implementations have some provisions for backwards compliance with older application-specific issues. Since not every application is likely or expected to support all these levels, it might be better to properly convert the formulas in your spreadsheets so they don't depend on such backward compliance features.

16.3 Problems converting spreadsheets to ODF

ODF 1.2 has been extensively tested by many experts from a variety of backgrounds. Conversion problems are most likely to be the result of historical product features being improperly mapped to the ODF standard.

Software sometimes strips out some functionality during the conversion. For instance Excel spreadsheets converted into OpenDocument Format 1.1 using Microsoft Office may not display drop-down lists. Some of the differences between historical application defined formats are small but can still cause significant problems.

A few are highlighted in the annotated version of the OpenFormula specification, which also provides useful background information.

Some vendors provide an overview of their design choices when implementing ODF; these notes (such as the implementation of ODF in Microsoft Excel) can provide additional clues. If you experience problems with conversion, follow these steps.

1. Test a more recent software version to see if the issue has been addressed.
2. Check if there are plugins or add-ins that might solve the problem.
3. Ask your vendor why it doesn't work and ask when the function can be restored.
4. If you can't wait, consider getting support to improve the product or build a plugin or add-on.
5. Alternatively, select a different desktop publishing tool that does meet your needs.

16.4 Known issues

16.4.1 Precedence issue

To give accurate outcomes to formulas, spreadsheets need to assign precedence (priority) to different mathematical operators. Precedence varies between products, which can cause issues.

In OpenFormula, prefix ‘-’ is given a higher precedence than ‘^’, to accommodate that ‘-2^2’ remains ‘4’ in Microsoft Excel and products that aimed to be compatible with it, such as OpenOffice, LibreOffice and Gnumeric. Because prefix ‘-’ had a lower precedence in Lotus 1-2-3, Quattro Pro, and Excel’s own Visual Basic, these products need to insert and remove parentheses when reading/writing expressions in OpenFormula where this matters.

16.4.2 Decimal separators

In a formula, to distinguish between subsequent parameters, OpenFormula uses the semicolon ‘;’. Many locations that don’t use US English use the comma ‘,’ as the decimal separator. Using the semicolon as the parameter separator eliminates confusion and the risk of incorrect implementation. Excel traditionally uses the comma as the function parameter separator, which may cause incorrect usage.

16.4.3 Empty parameters

In normal circumstances, OpenFormula does not allow empty parameters in formulas. By setting the application to behave according to the Basic level of implementation, that application can emulate the behaviour of Microsoft Excel to accept empty parameters in any position. Typical implementations will have many built-in functions, and most implementations also support one or more ways to create user-defined functions.

16.4.4 Workarounds to Excel AND/ OR bug

Microsoft Excel’s AND and OR functions historically did not work correctly in array formulas. This is a known bug. OpenFormula provides a correct definition of these functions. The traditional workarounds as recommended by some literature (using * and + inside array formulas as a work around) should however still work in ODF.

16.4.5 Intersection operators

In Microsoft Excel’s display format, the space is used as the intersection operator. This can easily lead to mistakes or be confusing at the least. The OpenFormula exchange format uses ‘!’ instead for intersection.

17 Support and training

Training and support options for the different users of ODF in your organisation.

17.1 Train users of office productivity tools

There are potentially 3 different user groups for Open Document Format (ODF) in your organisation, each with different needs in terms of support and training:

- users of office productivity tools
- technology teams who need to migrate from tools based on proprietary formats
- developers who are integrating or extending software based on ODF

Most office document applications function in a similar way. For example, word processor applications will likely have many features in common. The skills to use one will usually be enough to use another.

Most commonly used tools can also already create and save documents in ODF and can be set to do so by default.

So moving to ODF doesn't necessarily mean you need to provide extensive training.

17.1.1 Identify what your users need

Before you create a new training plan you should identify how much your users know or can find easily without help.

Tell your users as early as possible that they'll be using a new format. Teach staff how to check for ODF versions and how to check their documents before sending.

Spreadsheet users should be aware that spreadsheets converted to ODF version 1.1 rather than 1.2 will probably be missing formulas.

17.1.2 Training if you replace your existing tools

If you're going to upgrade or replace your tools, diagnostic testing or user research could tell you how much your users know about the new options. To train them, you could plan one or more of:

- online, self-supported study (usually low cost)
- using material from commercial publishers
- using open content you can customise, rewrite and republish
- support from super users - a network of experts
- training provided by a third party

17.1.3 Formal qualifications for users

You might consider encouraging some users to take formal qualifications as part of their personal development plans.

In the UK, the [e-skills Sector Skills Council](#) manages [ITQ](#), the national vocational qualification for IT users. Most providers of the ITQ focus on learning units related to office software. You should choose a provider working with an official awarding body such as [The Learning Machine Ltd \(TLM\)](#), which publishes its materials online as open content. TLM is accredited as an awarding body by Ofqual in England and the Department for Education and Skills in Wales.

Members of the UK's [Open Source Consortium](#) provide training that leads to TLM qualifications. These are official qualifications that use well-known open source ODF implementations such as LibreOffice or Apache OpenOffice as a starting point.

17.2 Training and support for technology teams

Many organisations have been exclusively attached to a specific vendor and its applications and platforms, in some cases for decades. If this is the case in your organisation and those products cannot work with ODF documents or be adapted to do so, you'll need to plan for change.

Your technology team may have received training relating only to a single product or supplier ecosystem. Some may have careers that are directly linked to proprietary certification schemes. They may perceive a move to ODF as threatening their career prospects and this can become a blocker. You need to prevent or modify these perceptions in order to make the most of your organisation's move to ODF.

17.2.1 Building your team's capability

You should present the transition to ODF as an opportunity for your staff to broaden their knowledge and increase their skills. Approach this as a positive challenge for everyone involved.

You might want to consider:

- offering training to your existing team
- encouraging and rewarding those who are enthusiastic about the change and getting them to lead on the implementation of ODF
- recruiting new team members who can pass on their knowledge to your existing team

17.2.2 Bringing in skills from outside

You might want to find organisations that can provide strategic advice on how to successfully move your organisation away from locked-in, proprietary software. You should look for experts who have broad expertise and are not linked to specific vendors.

You also might need to bring in knowledge, skills and experience if there are gaps in your current team. You could:

- use freelance experts or companies (local SMEs or larger companies)
- set up a cooperative agreement with a similar organisation
- hire experienced staff to work within your organisation - any external, temporary capacity you hire should help to build internal capacity
- allow people within your organisation to develop the right skills outside your organisation (eg through secondments)

Be vocal about what you do. If a team member has gone on a course or a secondment, ask for a write-up or a presentation that can be shared with technology colleagues across government.

17.3 Support, service and development

17.3.1 Vendor-owned solutions

Support levels and arrangements will vary for ODF applications that are exclusively owned or operated by a specific company.

Vendors often have a network of official resellers, integrators, service providers and freelancers they work with. They help to configure software, provide training, and provide phone support.

If a company is unwilling to provide support when you need it, there may be alternatives.

Add-ons could be available for the vendor's product, for example. If you need specific ODF functionality in Microsoft Office, for instance, there are existing open source add-ons that provide support for track changes, font embedding, ODF fallback mechanism, form controls and other standard ODF features.

These solutions won't be available when the source code of the product is unavailable, or application integration has been refused by the vendor.

17.3.2 Open source solutions

For an open source project, there are no restrictions on who can provide support, service or training. There are no fixed minimum rates, and people can work for you directly.

If you have a long-term development or training need, you might consider employing your own developers or trainers. This might be possible for a fraction of the licensing cost you're paying for software. You'll build capability within your organisation and you'll be in a better, more informed position to deal with vendors.

Your organisation can choose to work with:

- individual experts on a freelance basis
- local SMEs such as [Collabora](#) or [Canonical](#)
- domain specialists such as [MultiRáció](#) or [KO GmbH](#)
- global companies such as [IBM](#) or [RedHat](#)

- Apache OpenOffice provides a list of [experienced partners](#) you could consult. LibreOffice has an [accreditation scheme](#), acknowledging certified developers and migration professionals, professional trainers and support professionals.

If you want to develop a solution using open source but aren't sure where to start you can:

- ask peers for advice and guidance
- search for advanced users who can help out
- join up with another organisation in a similar situation to combine funding and share expertise
- ask community experts for advice

17.4 Support for using developers with ODF

Code from open source applications is particularly flexible for reuse by developers, who can use it to create exactly what your organisation needs. Your organisation also might consider getting involved with communities working on projects that are helping you provide better service to your users. This can involve contributing financial or human resource, although not all communities are equally open to external participation. An organisation can also state its user requirements, make a budget available and outsource everything.

Different models for development also have different financial implications. You should take into consideration the potentially large difference in cost between:

- purchasing a licence for every user in your organisation
- sponsoring a feature
- hiring a developer as a freelancer
- allocating a staff member to develop what you need One benefit of the last option is that as your organisation's knowledge, skills and experience increase, so does the amount of control you have over the conditions, cost effectiveness and security of your IT.

17.4.1 Open source

[Open Source Consortium \(OSC\)](#), the UK's open source industry association, sponsors conferences and other events related to open source.

You can contact them for help finding recruiters that can match your needs with specialists who have a background in open source. When dealing with open source it's advisable to recruit people familiar with it, and to ask people with a strong open source background for advice with hiring. Many such experts are available through SMEs.

[Openforum Europe](#) maintains an open source library that includes information and guidance on managing issues that arise with open source rather than proprietary solutions, and government open source policies.

17.5 Helpful organisations, tools and events

There are no official support services for standards, but the OpenDoc Society offers tools such as [Officeshots](#) and [ODF Validator](#) to the user and developer community.

The OpenDoc Society also organises events such as the [ODF Plugfests](#) and provides resources such as programming recipes for developers and the website [opendocumentformat.org](#). They're able to offer limited support to organisations moving to ODF, for instance by helping them connect with technology providers and open source communities.

If you want to learn more about the standard, or are interested in specific subjects such as accessibility or advanced collaboration, you should consider joining the [OASIS ODF Technical Committee](#).

Open Document Format is also an international standard. Your national standards body (eg BSI in the UK, NEN in the Netherlands, ABNT in Brazil or GOST in Russia) might be able to provide training.

17.6 Support on the development of the standard

If you'd like to make a suggestion regarding or have a question about the ODF standard itself you can:

- contact members of the [OASIS ODF Technical Committee](#), which includes freelance technical ODF experts
- email the OASIS dedicated public mailing list
- contact your national representative on the international standards board (ISO/IEC JTC1 SC34)

ODF is built on the XML standard, so you can also contact an organisation such as [XML Guild](#) that brings together many high level XML and standards experts from around the world.

18 Overview of productivity software

A non-exhaustive list of productivity solutions that allow you to work with ODF, including product features and other important details.

18.1 Auxiliary products and services

After assessing the needs of your users, you'll understand what functionality your organisation and audiences require from office document handling solutions.

A variety of applications work with ODF 1.2 out of the box. You also might be able to continue using an existing application by using a plugin, add-on or extension.

Open source solutions allow cost-effective customisation to your organisation's needs. The sooner your organisation moves to support the ODF 1.2 standard, the better able you are to find the best-value solutions for your needs.

This chapter is a non-exhaustive list of productivity solutions that allow you to work with Open Document Format (ODF). Other solutions that could be relevant include:

- web viewers (eg [ViewerJS](#)) that allow you to make documents accessible on your website
- software libraries that allow automated document access (eg Apache ODF Toolkit)
- metadata removal software, eg:
 - [Datadistiller](#)
 - [MetaClean](#)
 - [Metadata Assistant](#)
 - Detailed discussion of such auxiliary products, projects and services is out of scope of this document.

18.2 AbiWord

[AbiWord](#) is a fast, lightweight word processor that can even run on older computers without much processing power. Features include real-time collaboration, including through a direct connection between computers, which offers more security. Its feature set may not be as wide-ranging as some others on the market, but it offers what the average user needs in a user-friendly solution.

- Current release: 3.0.1 (2014)
- Release schedule: feature driven
- Type: text editor
- Platforms: Linux, UNIX, Windows, Minix, ReactOS, QNX
- Customisation allowed: yes, entirely open source
- Supported ODF-version: 1.2
- Font embedding: no
- Change tracking support: (temporarily not available)
- Unique features:
 - private collaborative editing
 - extensive RDF metadata support
 - low hardware requirements

18.3 Apache OpenOffice

[Apache OpenOffice](#) is a community-driven open source productivity suite available on many different operating systems. It includes a word processor (Writer), a spreadsheet application (Calc), a presentation engine (Impress), a drawing and flowcharting application (Draw), a database and database front-end (Base) and

mathematics editing (Math). The software looks and feels familiar, with many reporting it's instantly usable if they've used other common office products.

Apache OpenOffice was conceived and designed as a complete office package, with different components that share the same elements, making for an easier learning curve for users. At the end of 2014 over 130 million copies of Apache OpenOffice had been downloaded. Future versions will be signed in a way compatible with Windows security settings, to allow safe and seamless installation. Bespoke releases are available from multiple parties.

- Current release: 4.4.1 (Feb, 2015)
- Release schedule: twice a year, custom releases
- Type: office suite
- Platforms: Linux, BSD, Windows, Mac OS X
- Customisation allowed: yes, entirely open source
- Supported ODF-version: 1.2
- Font embedding: no
- Change tracking support: yes
- Unique features:
 - hundreds of extensions available
 - interactive image Crop Feature
 - iAccessible2 accessibility framework
 - SVG support

18.4 Bread

[Bread](#) is a fully responsive HTML5-based collaborative document editing platform that runs in the browser, without any plugins/ActiveX/Flash/Java applets. The client is supported by all major browsers, devices, tablets and mobile phones. The server part of Bread can be installed on any Windows or Linux OS, and is available as a hosted solution as well.

The web client can be embedded within any application, making it a full document editing software and co-authoring solution. Bread consists of a word processor (Docs) and a spreadsheet application (Sheets).

- Current release: 1.2 (2015)
- Release schedule: Major release every year, minor release every 4-6 months.
- Type: text editor, spreadsheet
- Platforms: Browser (platform agnostic)
- Customisation allowed: no
- Supported ODF-version: 1.2
- Font embedding: no (announced for version 1.3)
- Change tracking support: no (announced for version 2.0)
- Unique features:
 - built-in automated versioning
 - real-time collaborative editing
 - available as both on-premise and in-cloud

- integrated chat
- share and embed document in webpages and public locations
- full API for easy integration
- built-in rights management to invite users with different edit rights
- ability to load documents in read only mode

18.5 Calligra Suite

[Calligra Suite](#) is a relatively new, open source office suite for desktops and tablets that has a unique user interface. It bundles a number of classical tools such as a text editor (Words), a spreadsheet (Sheets), presentation software (Stage), database (Kexi) with the other free creative tools inside the Calligra suite such as Krita, a digital painting application, and a number of desktop publishing features.

Calligra aims to be a powerful and easy-to-use tool for a wide audience rather than specialist needs. It's produced with the support of the KDE technology team, one of the largest open source developer communities in the world. Calligra is very modular and is built around the open source Calligra Office Engine, which contains many freely reusable components. These can be used to embed any of Calligra's capabilities inside bespoke applications.

Calligra recently started a new series of applications called Calligra Gemini, a touch-friendly tablet app version of Calligra's word processor and presentation application.

- Current release: 2.9.7 (Aug, 2015)
- Release schedule: monthly updates, bi-annual release
- Type: office suite
- Platforms: Linux, BSD (preliminary support for Windows, Mac OS X)
- Customisation allowed: yes, entirely open source
- Supported ODF-version: 1.2
- Font embedding: yes
- Change tracking support: (temporarily not available)
- Unique features:
 - strong RDF metadata support
 - ebook editor (Calligra Author)
 - creative whiteboards (Calligra Braindump)
 - touch-friendly tablet versions (Calligra Gemini series)
 - support for Google Docs, so you can link to and open Google documents for editing on your desktop.
 - integrated globe
 - music notation editor

18.6 EtherCalc

[EtherCalc](#) is a collaborative web spreadsheet that allows multiple users to do real-time editing on a shared spreadsheet. A free software project led by software

engineer Audrey Tang, it's based on the open source javascript spreadsheet engine SocialCalc. SocialCalc was created by a team led by Dan Bricklin, author in 1978 of the world's first spreadsheet Visicalc and member of the ODF TC Formula subcommittee that drafted the ODF 1.2 standard.

Inspired by the wiki model, EtherCalc aims to combine the authoring ease and multi-person editing of wikis with the familiar visual formatting and calculating capabilities of spreadsheets.

- Current release: continuous releases
- Type: spreadsheet
- Platforms: Web application, can be self-hosted on any OS
- Customisation allowed: yes, entirely open source
- Supported ODF version: 1.2
- Font embedding: not yet
- Change tracking support: provides an audit trail
- Unique features:
 - private collaborative editing when self-hosted
 - no setup required for hosted version, runs inside the browser

You can read about the creation and design of EtherCalc in the free online book [‘The Architecture of Open Source Applications’](#).

18.7 EuroOffice Professional

[EuroOffice](#) is an integrated office suite running on both Windows and Linux systems. It's an open source project that contains a text editor, spreadsheet, presentation software, database and drawing and other tools. EuroOffice is built on an enhanced code incorporating code from OpenOffice.org, LibreOffice and Apache Open Office, as well as joint R&D work of the company MultiRáció and Szeged University in Hungary.

EuroOffice 2015 Professional includes technical support and a number of extensions, including various language tools, a real-time dictionary, professional map charting tools, mathematical solvers for linear, quadratic, non-linear and Hungarian algorithm optimization problems. EuroOffice also contains a custom extension creator.

- Current release: EuroOffice 2015
- Release schedule: annual
- Type: office suite
- Platforms: Windows, Linux,
- Customisation allowed: limited to core platform
- Supported ODF-version: 1.2
- Font embedding: yes
- Change tracking support: yes
- Unique features:
 - adaptive interface that changes the size of menu items according to usage frequency

- many different plugins for sparklines, statistics modelling, physics simulation, mail archiving, linguistic, mathematical and map charting tools, measuring writing progress
- ExtensionCreator Pro

18.8 Gnumeric

[Gnumeric](#) is a versatile and powerful spreadsheet, created and maintained by the open source [GNOME](#) project. Its stated advantages are that it:

- is fast and stable
- uses an intuitive interface that users familiar with similar applications will find easy to pick up
- can be used in a wide variety of environments
- has a wealth of features and is particularly reliable for the accuracy of its calculations
- includes a highly configurable data importer and exporter which allows many formats to be added as plugin software routines

The core architecture is designed to ensure that Gnumeric can comfortably scale to moderately large loads (1 million cells) while remaining usable on older hardware.

- Current release: Gnumeric 1.12.23 (Aug 2015)
- Release schedule: feature driven
- Type: spreadsheet
- Platforms: Linux, Mac OS X, BSD (Windows version is possible, but needs a sponsor)
- Customisation allowed: yes, entirely open source
- Supported ODF version: 1.2
- Font embedding: not yet
- Change tracking support: not yet
- Unique features:
 - Unique analytics capabilities
 - Low hardware requirements
 - Offers programmatic utilities for instance for automated comparing and converting of spreadsheets

18.9 Google Apps

[Google Apps](#) includes a collaborative web text editor (Docs) and spreadsheet application (Sheets). Both products allow multiple users to do real-time editing on a shared document and can be used through dedicated apps running on 2 mobile platforms:

- Google's officially branded version of Android

- Apple iOS devices.

Google Apps is a commercial online application that users must register to use. Instead of track changes, Google Docs lets users make suggestions that the owner of a document can accept or reject later. These ‘Suggested Edits’ are available for anyone with commenting access, but cannot be exported for use in another application.

- Current release: (no version published, continuous releases)
- Type: text editor, spreadsheet
- Platforms: Web application, supports most browsers
- Customisation allowed: no Supported ODF version: import ODF 1.1/1.2, export ODF 1.1
- Font embedding: not yet
- Change tracking support: provides history and ‘suggested edits’, cannot be exported to other applications
- Unique features:
 - offline editing and remerging capabilities
 - no setup required for browser version, installable app version available as well
 - offers a number of APIs (application programming interfaces) for accessing documents programmatically
 - online Google forms offer convenient data collection in online spreadsheets

18.10 Kotype

[Kotype](#) is an open source collaborative editor that has both a downloadable application version and a [web-based version](#).

Part of the WebODF family of open source ODF tools, Kotype allows multiple users to work simultaneously on the same document using any modern browser. It’s a clutter-free solution that’s been designed around the ODF standard to create high quality and fully interoperable ODF documents.

- Current release: 0.9 (2015)
- Release schedule: unknown
- Type: text editor (spreadsheets and presentations can be viewed, not yet edited)
- Platforms: browser based
- Customisation allowed: yes
- Supported ODF-version: 1.2
- Font embedding: yes
- Change tracking support: no
- Unique features:
 - lossless round-trip editing for ODF text documents
 - collaborative editing
 - server can be installed on site

18.11 LibreOffice

[LibreOffice](#) is a community-driven open source productivity suite available on a variety of operating systems. Developed with the support of the not-for-profit Document Foundation, LibreOffice includes a word processor (Writer), a spreadsheet application (Calc), a presentation engine (Impress), a drawing and flowcharting application (Draw), a database and database frontend (Base) and mathematics editing (Math).

The suite fits into existing IT environments and works with Microsoft SMS Server, Novell ZENWorks, IBM Tivoli, Citrix, etc. Custom versions are available with enterprise in mind, offering features such as MSP patch files and Group Policy Object management. There are official releases twice a year, with bespoke releases available from multiple parties.

- Current release: 5.0.1 (Aug, 2015)
- Release schedule: 2 releases per year
- Type: office suite
- Platforms: Linux, BSD, Windows, Mac OS X
- Customisation allowed: yes, entirely open source
- Supported ODF-version: 1.2
- Font embedding: yes
- Change tracking support: yes
- Unique features:
 - Microsoft Visio reader/viewer and import support
 - Google Docs compatible (import/export, read/write)
 - CMIS protocol support to access Document Management Systems
 - rich ecosystem of hundreds of extensions
 - iAccessible2 accessibility framework

For a more comprehensive list see the project's own [feature comparison with Microsoft Office 2013](#).

18.12 Microsoft Office

Microsoft Office includes presentation software (Powerpoint), a spreadsheet (Excel), a word processor (Word), a database (Access) and a number of smaller applications. Office offers an extensive feature set that closely integrates with other products from Microsoft as well as third party vendors such as Dropbox. It offers a variety of pricing plans, subscription and licensing models, as well as on-premise and cloud-based deployment options. Enterprise deployment support is also available.

New releases of the on-premise version are made roughly every 3 years. However, with the introduction of Office Online, enhancements are released on an ongoing basis. See the [Office Online roadmap](#) for further details.

Office 2013 for Windows supports ODF 1.2, but without change tracking.

Office 365 supports ODF 1.2 and cloud collaboration capabilities but does not include change tracking for exported documents.

- Current release: ‘15’ / Office 2013 (Feb, 2013), Office 365
- Release schedule: new release per 3 years, or ongoing for [cloud version](#)
- Type: office suite
- Platforms: Windows, plus cross platform browser support for Office Online and mobile clients available for iOS and Android.
- Customisation allowed: no
- Supported ODF-version: 1.2
- Font embedding: no
- Change tracking support: no
- Unique features:
 - convenience functions such as Quick Analysis, Flash fill and Chart recommendations
 - SmartArt diagrams
 - Ink annotations
 - editing of embedded videos
 - includes auxiliary applications such as email and note-taking (no support for ODF, only legacy formats)
 - hosted web version of apps with collaborative editing, including messaging
 - digital signatures without need to install plugin

Microsoft Office has official support for ODF 1.1 in Office 2007 for Windows (as of Service Pack 2), Office 2010 and Office 2013 (read only for ODF 1.1, read and write for ODF 1.2). Users of versions of Office from Office 2003 can also use the <http://odf-converter.sourceforge.net/>, which supports ODF 1.1 and has basic change tracking support.

None of the released versions of Microsoft Office for Apple OS X have had support for OpenDocument Format at the time of publication.

18.13 Owncloud Documents

[Owncloud](#) is a suite of client-server software for creating file hosting services and using them. Documents is a lightweight editing solution that can conveniently be used from any device, to quickly view and perform basic edits to ODF documents. Owncloud Documents is part of the WebODF family of open source ODF tools.

- Current release: 8.1 (2015)
- Release schedule: annual
- Type: text editor
- Platforms: browser based
- Customisation allowed: yes
- Supported ODF-version: 1.2
- Font embedding: yes
- Change tracking support: no
- Unique features:
 - lossless round-trip editing for ODF text documents
 - document management

18.14 OX Text

[OX Text](#) is an open source collaborative web text editing solution that allows multiple users to do real-time editing on a shared document. OX Text is part of a modular cloud based solution that also offers open source spreadsheets, a calendar, address book, mail, messaging, storage and time management. It offers centralised document sharing with version control inside the browser.

OX Text backend is licensed under a GPLv2 license and frontend under a CC BY-SA-NC license (see the FAQ). Commercial support is available. A subscription-based hosted version is also available as a white label. Track changes information is currently limited to legacy formats and cannot be imported or exported for use in another ODF application.

- Current release: 7.6.2 (2015)
- Release schedule: twice a year
- Type: text editor, spreadsheets
- Platforms: Browser-based (see the system requirements)
- Customisation allowed: yes
- Supported ODF-version: 1.2
- Font embedding: no
- Change tracking support: only legacy formats
- Unique features:
 - lossless round-trip editing for ODF documents
 - collaborative editing with exclusive editing rights
 - re-use paragraph and table styles imported from the original ODF documents
 - auto-save feature saves all document changes automatically

18.15 Wodo.TextEditor component

[Wodo.TextEditor](#) is a lightweight and generic OpenDocument Text editor component that can be added to a website or mobile or desktop application using a few lines of JavaScript. It allows users to quickly view and perform basic edits to ODF documents from any device that lacks ODF support. Just open the webpage containing Wodo, select your file from the toolbar and start editing. Wodo.TextEditor works directly in the browser without any dependencies, so there's no need for any server support. Wodo.TextEditor is part of the WebODF family of open source ODF tools.

It is open source so you can install it anywhere and customise it to fit any purpose. The barebones Wodo is fully functional and usable even on the public demo.

- Current release: 0.5.9 (2015)
- Release schedule: feature driven
- Type: text editor (spreadsheets and presentations can be viewed, but not yet edited)
- Platforms: browser based

- Customisation allowed: yes
- Supported ODF-version: 1.2
- Font embedding: yes
- Change tracking support: no
- Unique features:
 - lossless round-trip editing for ODF text documents
 - easy integration with websites and mobile/desktop applications
 - documents do not leave your computer while editing

19 Costs and benefits of ODF

Useful information for building a business case for moving your organisation to ODF.

19.1 Benefits of moving to ODF

Open standards enable suppliers to compete on a level playing field. This competition is leading to innovative new solutions, [improved processes](#) and lower costs.

Open Documents Format (ODF) offers many benefits, including:

- lower ICT costs
- increased flexibility
- better ICT governance
- making it easier to share documents across different software for editing
- making it possible to manage intensely cross-linked documents, such as legal texts and legislation
- allowing multiple output sources, eg to a website, to a printer, or to specialised accessibility equipment
- preventing problems with formatting
- allowing much stricter security checks on incoming and outgoing documents to prevent common cyber-attack scenarios
- better long-term preservation of information
- providing more choice of applications (the Wikipedia page on OpenDocument lists 26 office software applications supporting or partly supporting ODF)
- no need for plugins or converters in open source solutions like Apache OpenOffice and LibreOffice, where ODF is the default

19.2 Cost savings by European governments

The following are mainly examples of savings that have been made as a result of moving to both ODF and open source software. The selection of an open standard will stimulate competition for government contracts and help to decrease costs for government of office productivity tools.

Total cost of ownership (including exit costs) must continue to be considered by government organisations on a case by case basis. In the same way, transition costs from current technology to meet ODF requirements cannot be included in cost assessments, as they represent the cost of vendor lock-in.

19.2.1 France

MIMO, an inter-ministerial working group, is implementing ODF native software solutions on the workstations of 11 of France's 17 ministries. The LibreOffice office suite is now [installed on nearly all 500,000 desktops](#) of France's ministries:

- Energy (Ecology)
- Defence (Défense)
- Interior (Intérieur)
- Economy
- Justice,
- Agriculture
- Culture and Communication
- Education
- Finance
- Health and Social Affairs
- Foreign Affairs

None of these departments need licences, meaning none pay for them.

Toulouse, France's fourth largest city, has saved €1 million by migrating all its desktops to LibreOffice. [A study](#) published by the European Commission's Open Source Observatory and Repository (OSOR) notes that France demonstrates the largest example of a [public administration using open source on workstations](#) in Europe. The country's Gendarmerie (a branch of the French armed forces) has Ubuntu Linux and LibreOffice running on 72,000 workstations (including the outposts in French Polynesia).

According to Major (Commandant) Stéphane Dumond this approach lowers the total costs of workstations by 40%, a combination of lower licence costs, much easier and central IT management and a huge decrease in the number of local technical interventions. "The decrease in licence costs are only the tip of the iceberg", Dumond [told South Korean government executives](#) in 2014.

19.2.2 Italy

Free software such as LibreOffice and OpenOffice is [popular in the public sector](#) in Italy. The Italian city of Trieste [expects to save €900,000 in proprietary office software licences](#) between 2014 and 2017 by moving to the Apache OpenOffice suite. The administration of the Italian region of Emilia-Romagna is expecting savings of €2 million by moving to free software.

19.2.3 Spain

In Spain several regions are switching or have switched to free software office suites. Examples include [the administration of the Spanish autonomous region of Valencia](#). In 2013 it moved 120,000 desktop PCs of the administration, including schools and courts, to LibreOffice. The migration will save the government €1.5 million per year on software licences.

“Apart from economic benefits, the commitment to free and open source software makes the solutions available in the Valencian language as well as in Spanish, and brings IT vendor independence, which encourages competition”, the ICT department’s director general, Sofia Bellés, said in a statement at the time. “We also have the freedom to modify and adapt the software to our needs.”

In Galicia, [around 1,000 workstations](#) used by the region’s public authorities had LibreOffice installed in 2014. The government also said it would start raising awareness among the region’s public administrations about the advantages of sharing and promoting the reuse of ICT solutions.

19.2.4 Germany

Munich [switched all of its 14,200 desktops](#) to Linux and LibreOffice.

19.2.5 The Netherlands

The annual ICT costs for the Dutch municipality of Ede are now 24% lower than its peers. An [OSOR study](#) states that “most of this reduction can be explained by Ede’s extremely low spend on software licenses: only €56 per full-time equivalent employee instead of €731. Such a large reduction was achieved by moving from proprietary to open source software.”

The municipality is running dozens of open source solutions, including LibreOffice.

19.2.6 Other municipalities

Further examples of municipalities making savings through open source are:

- Las Palmas
- Limerick
- Gummersbach
- Grygov
- Århus
- Arles
- Vieira do Minho
- Voreppe.