Logius
Ministerie van Binnenlandse Zaken en Koninkrijksrelaties

OpenDocument Format

The future of ODF

Jos van den Oever
Logius / KOOP
Ministery for the Interior
The Netherlands

Jos van den Oever

Ministery of the Interior

The Netherlands

WebODF

OASIS
Advancing open standards for the information society

What is the point of ODF?

application-independent and platform-independent digital document file format

Real choice for computer users

Computer user activities in 2015:

- *Read/write email*
- *Read/write web*
- *Read/write office documents*

What is OASIS?

- previously SGML Open
- DITA, DocBook, XLIFF
- ebXML (with the United Nations)
- Legal XML (Akomo Ntoso)

OASIS
Advancing open standards for the information society

W3C®

ISO International Organization for Standardization

Logius joins OASIS

- Logius provides ICT infrastructure for the Dutch government
- 2 years, 16 hours a week to work on ODF standard
- Government takes a leading role in essential file formats

How is the standard made?

- Issue database
- Biweekly phone conference
- Public comment list
- Coming soon: version control
- Prototypes
- Sync to ISO

OASIS
Advancing open standards for the information society

Version history of ODF

- 1.0, May 2005, 706 pages
  - application-independent and platform-independent digital document file format
- 1.1, Feb 2007, 738 pages
  - soft page breaks, navigation order for slides, alt texts
- 1.2, Sep 2011, 1115 pages
  - digital signatures, RDF based metadata, OpenFormula, tables in presentations, database front end documents

What is the ODF-TC working on?

Next version: ODF 1.3

- New features: mainly bugfixes
- +200 open issues https://issues.oasis-open.org/
- Change tracking subcommittee
- Moving LibreOffice/OpenOffice extensions into the standard

Specification is just the start

- Local communities
- Education
- Strict policy for adoption
- Migration
- Documentation
- Entrepeneurs
- Feedback

The future of ODF

- testing and certification
- profiles
- scripting
- real time change tracking
- upgrade / downgrade instructions
- HTML storage format
- normalization (ODF in Git, dogfooding ODF)
- standardize handling of erroneous files

- *Theme support*
- *your suggestions ...*

ODF Plugfests:
testing
ODF implemenations

ODF | Plugfest

XI

SEPT 15-16 2015, THE HAGUE (NL)

ELEVENTH EDITION

LibreOffice

ODF Plugfests

1	The Hague, NL	Jun, 2009
2	Orvieto, IT	Nov, 2009
3	Granada, ES	Apr, 2010
4	Brussels, BE	Oct, 2010
5	Maidenhead, UK	Feb, 2011
6	Berlin, DE	Jul, 2011
7	Gouda, NL	Nov, 2011
8	Brussels, BE	Apr, 2012
9	Berlin, DE	Oct, 2012
10	London, UK	Dec, 2014
11	The Hague, NL	Sep, 2015
12	...	Spring 2016

OdfAutoTests

- Provide a small snippet as input
- Run through a collection of office suites
- Summarize the output and show example renderings

odt	input	Calligra	Abiword	LibreOffice
valid ODF	true	false	true	true
styles.xml				
'ough-style)	true	true	false	true
tyle='wave'	true	true	false	false
pdf	input	Calligra	Abiword	LibreOffice
success		true	true	true
page 1		width: 595.0 height: 842.0 ~~hello world~~	width: 595.2756 height: 841.8898 ~~hello world~~	width: 595.0 height: 842.0 ~~hello world~~

http://autotests.opendocumentformat.org/

OdfAutoTests: join the team!

- ODF Plugfest Online
  - testing all year round
  - tie in with implementation continuous integration
  - web interface for writing and commenting on tests
  - live ODF implementation scoreboard: counting to 100%
- Open positions:
  - Front end programmers
  - Back end programmers
  - Officeshots operators
  - Test writers

http://autotests.opendocumentformat.org/

ODF profiles

- ~600 element names
- ~1300 attribute names
- Nobody implements all of them.

- Find out what elements are used in the real world
- Specify profiles: 'full text', 'markdown compatible', 'annotated open data', etc

Scripting + API

- in ODF 1.2, spreadsheets formulas were standardized

  before: `table:formula="oooc:=[.A2]+[.A1]"`

  after: `table:formula="of:=[.A2]+[.A1]"`

  before: `table:formula="oooc:=SUM([.B1:.B2])"`

  after: `table:formula="of:=SUM([.B1:.B2])"`

- SMIL (Synchronized Multimedia Integration Language)
  - slide transitions

- No standardized scripting language
  - `<office:scripts/>`

Change tracking

- In ODF 1.2: no change tracking for table cells in text

> "The <table:tracked-changes> element is usable within the following element: <office:spreadsheet>"

Real Time Change tracking

Work together with multiple people on one file at the same time

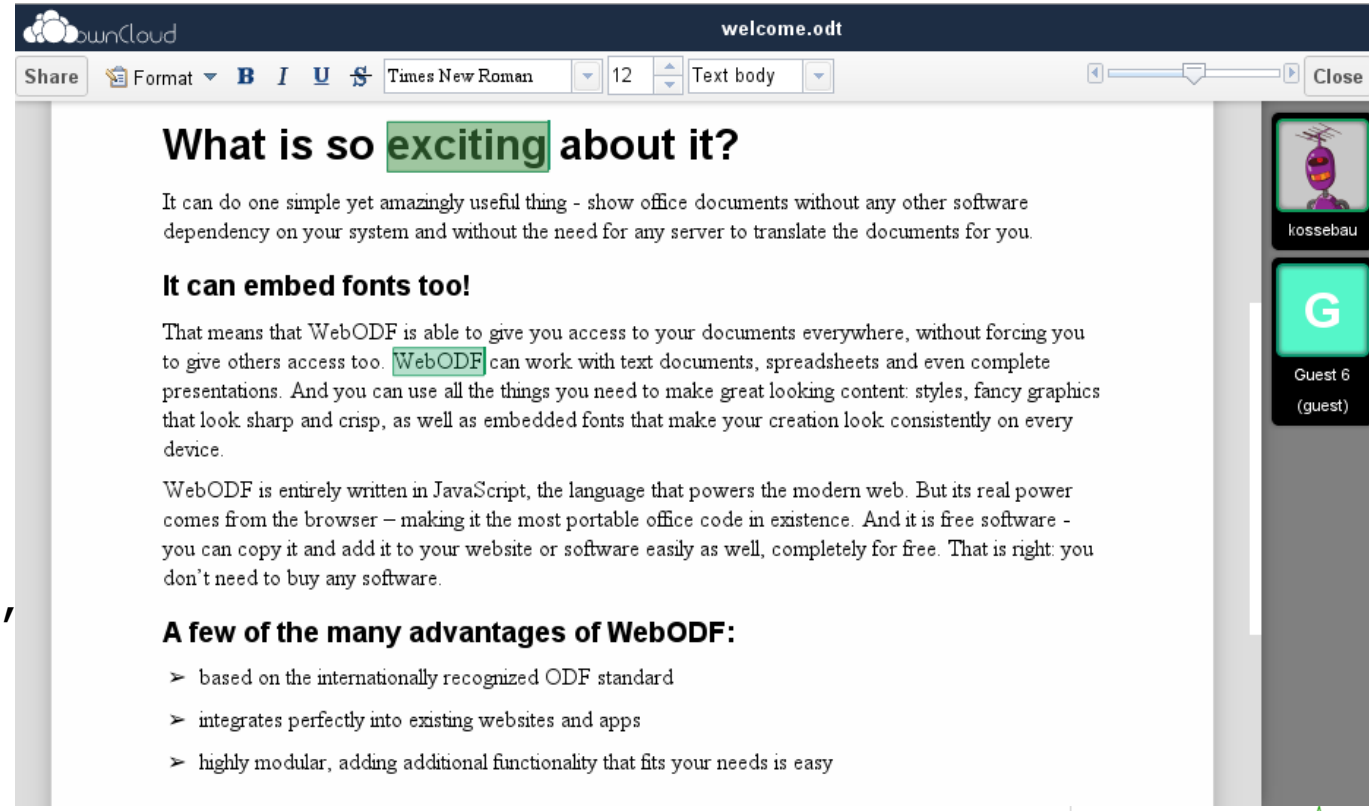
- detailed history
- subcommittee

- WebODF,

  OX Documents,

  MS Office Online,

  Abiword



Upgrade / downgrade instructions

New versions of the specification come with a list of changes

1.0 ↔ 1.1 ↔ 1.2 ↔ 1.2 extended ↔ 1.3 ↔ 1.3 extended

Implementations do not automagically support a new version

New implementations do not usually support writing to older versions

Should we standardize how to convert?

Should we provide a reference implementation?

Publishing ODF on the web

- Problem: many people do not have ODF support on their device

Solution 1:

- convering ODF to PDF with embedded ODF

Solution 2:

- convert ODF to HTML with embedded ODF

Solution 3:

- convert ODF to HTML with embedded ODF and editor

Comparing ODF and HTML

ODF	HTML
(paginated) text, spreadsheets, presentations, drawings	single long page of text
images, videos, drawings, math	images, videos, drawings, math
usually a single file	usually multiple files on multiple servers
created with office software	created by server software, vi or emacs
fixed versions	'living standard'

Developers dogfooding ODF

remove unneeded differences from between XML documents

- Code documentation in MarkDown README files
  - why not use README.odt?
- ODF implementations currently not Git-friendly
  - Saving ODF files gives changes in the file that are not related to the actual changes
- ODF TC is moving the specification to a source revision system

prototype for normalization at
https://gitlab.com/odfplugfest/odfhistory

Normalization of XML

remove unneeded differences from between XML documents

- UTF-8 encoding
- sort attributes `<a b="" a=""/>` → `<a a="" b=""/>`
- use double quotes `<a a='a'/>` → `<a a="a"/>`
- convert CDATA to text `<![CDATA[hello]]>` → `hello`
- indenting
- empy elements `<a>` → `<a></a>`

standardized as C18N at W3C

Normalization of ODF

remove unneeded differences from between ODF documents

- XML normalization
- standard namespace prefixes
- names and order of automatic styles
- sort automatic styles, entries in manifest, meta data
- remove layout specific information: page number
- add calculated values
- keep or remove empty elements
- normalize units: cm, pt, px

https://gitlab.com/odfplugfest/odfhistory/

Example diff for normalized ODF

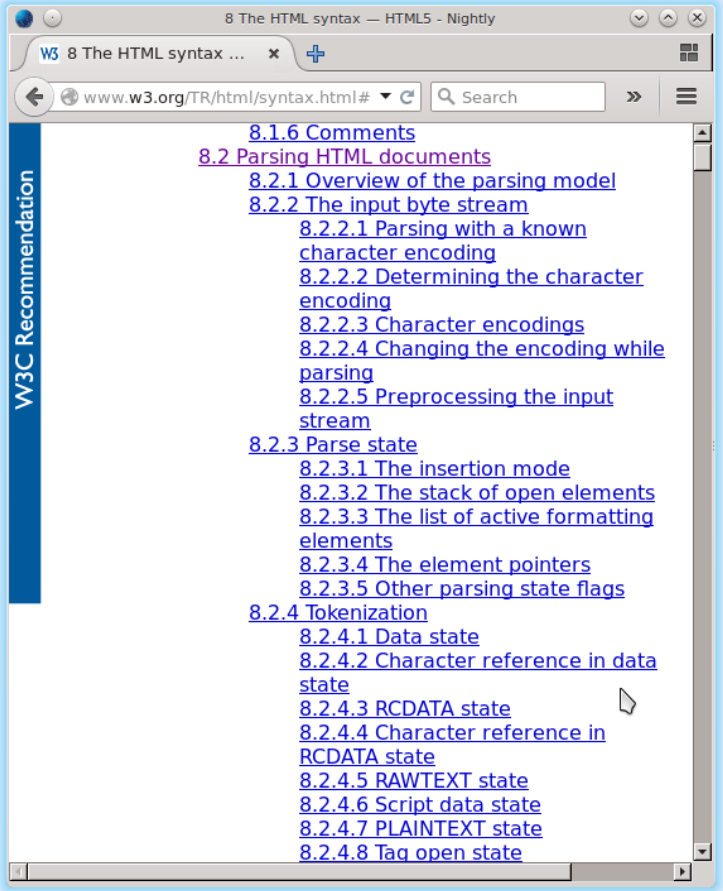
```
   <office:meta>
     <dc:creator>TC-Admin Editor</dc:creator>
-    <dc:date>2013-05-29T12:45:56.44</dc:date>
-    <dc:description>This is a modification of the ODF 1.1 OASIS S
+    <dc:date>2013-08-09T13:21:42.32</dc:date>
+    <dc:description>This is a modification of the ODF 1.1 OASIS S
 </dc:description>
     <dc:language>en-US</dc:language>
-    <dc:subject>Working Draft 07 of ODF 1.1 with Errata 01 applie
-    <dc:title>Open Document Format for Office Applications (OpenD
+    <dc:title>Open Document Format for Office Applications (OpenD
     <meta:creation-date>2003-02-05T12:39:40</meta:creation-date>
-    <meta:document-statistic meta:character-count="1258767" meta:
-    <meta:editing-cycles>1535</meta:editing-cycles>
-    <meta:editing-duration>P53DT7H48M11S</meta:editing-duration>
+    <meta:document-statistic meta:character-count="1258621" meta:
+    <meta:editing-cycles>1541</meta:editing-cycles>
+    <meta:editing-duration>P53DT8H7M31S</meta:editing-duration>
     <meta:generator>OpenOffice.org/3.4.1$Win32 OpenOffice.org_pro
-    <meta:keyword>Errata</meta:keyword>
-    <meta:keyword>OASIS</meta:keyword>
-    <meta:keyword>ODF</meta:keyword>
-    <meta:keyword>OpenDocument</meta:keyword>
     <meta:print-date>2004-09-17T15:13:31</meta:print-date>
     <meta:printed-by>Daniel Vogelheim</meta:printed-by>
```

more examples at
https://gitlab.com/odfplugfest/odfspecifcationhistory

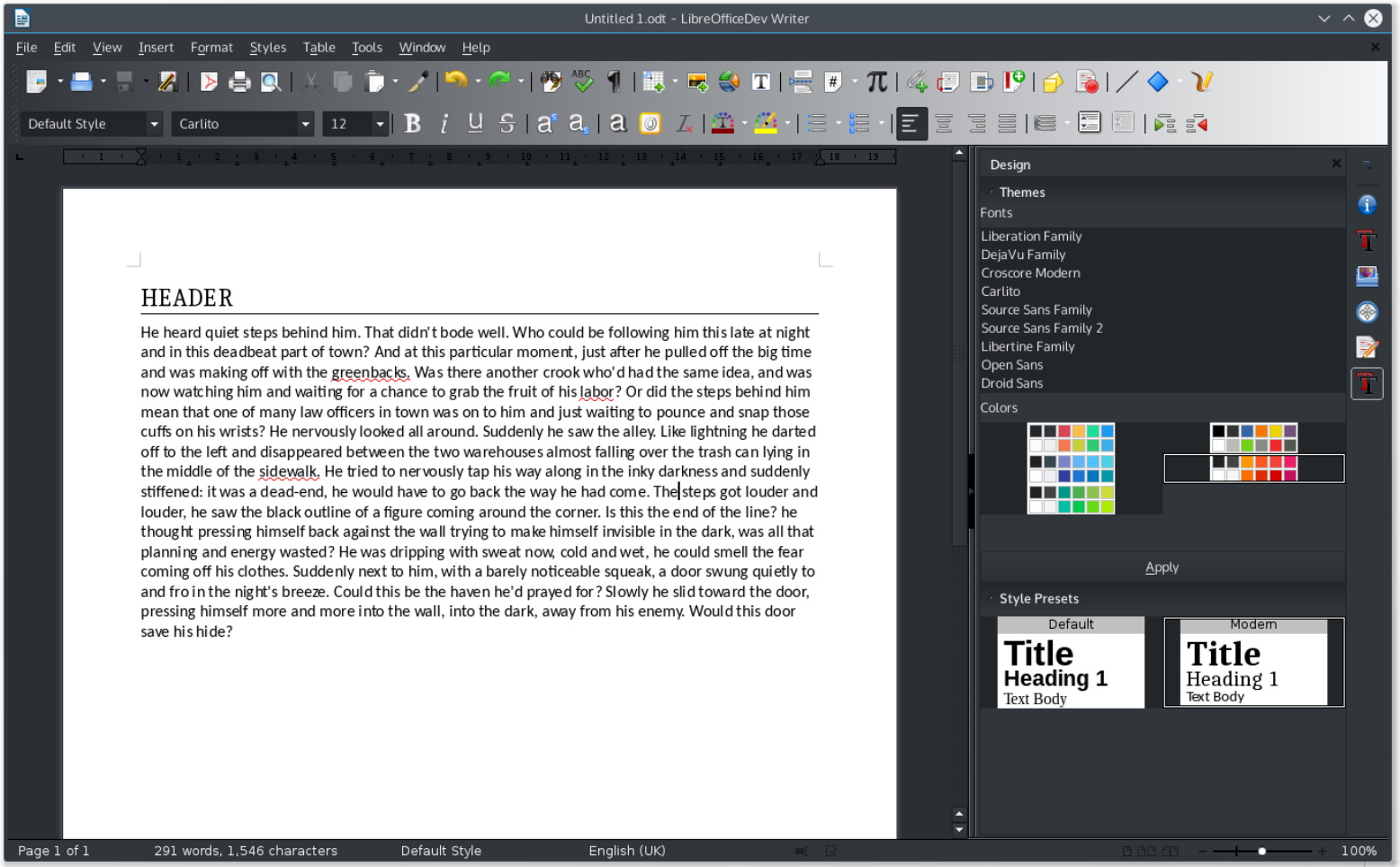
Standardize handling of erroneous files

- HTML ↔ XHTML
- Parsing HTML is a science

Should we formalize how to handle invalid files?



Theme support



The future of ODF

- testing and certification
- profiles
- scripting
- real time change tracking
- upgrade / downgrade instructions
- HTML storage format
- normalization (ODF in Git, dogfooding ODF)
- standardize handling of erroneous files

- *Theme support*
- *your suggestions ...*