OpenOffice.org XML File Format

David Hofert

Manager, XML Emerging Technologies Group

Sun microsystems

We make the net work.

Agenda

- Overview of OpenOffice XML File Format
- Details and Benefits
- Going Forward
- Questions

OpenOffice XML File Format

- Open Format Benefits
  - OpenSource reference application available from OpenOffice.org (OOo)

    http://xml.openoffice.org/
  - Developed under open source model
  - Fully documented, no “secret sauce”
  - Uses other open standards whenever possible
  - Eliminates dependencies on a single vendor

OpenOffice XML File Format

- Suitable for all Office Work
  - Supports text, spreadsheet, presentation, drawing, charts, formulas
- Suitable for Editing
  - Need a format that retains document structure
    - e.g. table of content must be updated
  - Output formats (e.g. PDF) only capture current state

OpenOffice XML File Format

- 1st Class XML Implementation
  - Native, Complete XML
    - Exceptions: images, OLE objects (as binary data)
    - No additional data necessary to view information
    - No 'hidden' data
  - Accessible through many third party products
    - XML editors, XML transformers, XML databases
    - DTD is provided with Office bundle

OpenOffice XML File Format

- Only One File Format – XML!
  - No 'lossy' exchange format, Native XML only
  - Complete document content
    - Complete content and style information – cleanly separated
  - Used by
    - OpenOffice.org 1.0, StarOffice 6.0,
      RedOffice 1.0, other OpenOffice.org-based suites

Other Standards Incorporated

- XHTML
  - Paragraph and heading structure, lists, tables
- SVG
  - Graphical elements
- XSL-FO
  - Many formatting attributes
- MathML
- XLink, Dublin Core

Document Details

- Document Root Contains Meta Information
  - Title, description, author, date (largely Dublin Core)
  - Settings (application specific), scripts and macros, font declarations
  - Style declarations
  - Automatic styles ('direct' formatting)
  - Master styles (pages)
  - Document body

Document Body

- Contains all Document Content
- Set Sequence of Content Elements
  - Paragraphs, headers, lists
  - Tables
  - Sections, indices
  - Graphics, frames, shapes
- Also Includes Some Specialized Elements
  - e.g. change tracking

Sample Body

```
<office:body>
 <text:p text:style-name="Standard">Hello</text:p>
 <text:p text:style-name="Standard">World</text:p>
 <table:table table:name="Table1"
              table:style-name="Table1">
  […]
  <table:table-row>
   <table:table-cell table:style-name="Table1.A1"
                     table:value-type="string">
    <text:p>Hello</text:p>
   </table:table-cell>
   <table:table-cell table:style-name="Table1.B1"
                     table:value-type="string">
    <text:p>World!</text:p>
   </table:table-cell>
  </table:table-row>
 </table:table>
 <text:p text:style-name="Standard"/>
</office:body>
```

Text

- Text Structure is HTML-Like
  - Paragraph, headers
    - Paragraph is basic text entity
    - <text:p> <text:h text:level="1">
  - Whitespace handling
    - Whitespace compression
    - Special elements <text:tab> <text:s>
  - Lists
    - <text:ordered-list> <text:unordered-list>
    - <text:list-header> <text:list-item>

Styles

- Two Types of Styles
  - User-defined
    - Styles used in paragraph
  - Automatic
    - 'Direct' formatting
    - E.g. user hits 'bold' button
- Exact Same Syntax

File Packaging

- Package Format
  - Combines several (XML & other) files into one
  - Provides efficient access to subdocuments
  - Necessary for efficient compound documents
  - Allows additional features
    - Compression – via ZIP packaging
    - Encryption – plain text obscured or can be hard encrypted

Benefits: Easy to Process

- Easy to Parse
  - Content & Presentation
    - Content for processing
    - Presentation for display
    - Process only what you need
  - Consistent
    - E.G. identical table model across applications
    - **One** format, consistent across applications

```
<table:table-cell
  table:formula=
    "=PI()"
  table:value-type=
    "float"
  table:value=
    "3.14159265358979">
<text:p>
3,14
</text:p>
</table:table-cell>

<text:date
  style:data-style-name=
    "N5079"
  text:date-value=
  "2002-07-10T15:22:22">
Mittwoch, 10. Juli 2002
</text:date>
```

Easy to Generate

- Minimal document has three elements
- Add additional information as needed
- Specify content and layout separately

```
<?xml version="1.0"
      encoding="UTF-8"?>
<office:document
  xmlns:office="..."
  xmlns:text="..."
  office:class="text"
  office:version="1.0">
<office:body>
<text:p>
Hello World
</text:p>
</office:body>
</office:document>
```

XML Transformation Capability

- Can Display Files Without Full Office
  - XSLT: OOo XML → HTML, WML
    - Files then viewable with XSLT + browser
    - Good quality, but not WYSIWYG
    - Aids document longevity (when archiving)
  - AxKit/Perl: OOo XML → HTML
    - uses DocBook-like intermediate format
  - XSLT: OOo XML ⇆ DocBook

More Transformation Benefits

- Use Filter to R/W other File Formats
  - Very useful for legacy data
  - Independent conversion filter creation costly
- XML-Based Filter Transformation
  - Use XML file format as intermediate
  - Simplifies development

Partial Viewing/Editing

- Binary formats
  - Require complete understanding
  - Can't just skip a few bytes
- XML-based format
  - Easily extract interesting information
  - Simplified document view
  - Extract simplified data for viewing on devices

Archiving and Indexing

- Archiving
  - What can we read in 50 years?
  - Often required, e.g. for public records
  - XML is plain text
  - Can be used without office
- Indexing
  - Search large data repository

Office as Layout Engine

- Generate Reports
  - Traditional method:
    - Somehow import data
    - Manually prepare charts, graphs, text format
  - New method: generate XML
    - Print or edit with OpenOffice.org
    - Separation of content and layout helps speed process
      - Preset styles, focus on content generation

An Extensible Format

- New Features Can Be Added
  - E.g. text grid added after OOo/SO release
- Extensibility Inherited from XML
  - Use of namespaces avoid name clashes
- Application Support
  - Tolerance for externally-generated attributes
  - Enables backwards compatibilty
  - Enables integration with document mgmt. systems

Future Developments

- Standardization
  - – Plans in place to take format to standards organization
- Continued Evolution of Format
  - – E.g. additional attributes for layout grid
- New Features Under Evaluation
  - – Digital signatures and encryption enablement

Summary

- OpenOffice.org XML Format Makes Sense
  - Native, complete XML leverages XML strengths (ascii, XSLT, structure, etc.)
  - Ensures longevity of data & files
  - Reduces cost of archiving, searching & indexing
  - Easy transition from native format to format suitable for web, web services, legacy apps.
  - Open, soon-to-be-standard format ensures you don't get bound to any specific application

David Hofert

david.hofert@sun.com

We make the net work.

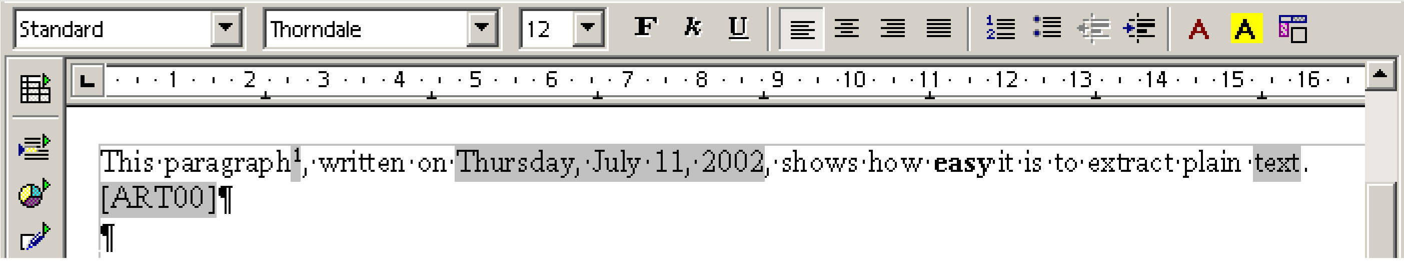
Background Slides

Easy to Process Example, 1 of 2

- Example: extract plain text
  - All text is contained in <text:p>, <text:h>
    - Always: text body, text sections, tables, frames
    - Contains characters + markup
      - Markup (fields) contain text representation
      - Exception for certain nested content
      - Footnotes, endnotes, annotations, frames
    - Result: 2+4 elements needed to extract plain text
  - Formatting in <text:span>

Easy to Process, 2 of 2

```
<text:p text:style-name="Standard">
This paragraph
<text:footnote text:id="ftn0">
<text:footnote-citation>1</text:footnote-citation>
<text:footnote-body>
<text:p […]>footnote</text:p></text:footnote-body>
</text:footnote>
, written on
<text:date […]>Thursday, July 11, 2002</text:date>
, shows how
<text:span text:style-name="T1">easy</text:span>
it is to extract plain
<text:bookmark-start […]/>text<text:bookmark-end […]/>.
<text:bibliography-mark […]>[ART00]</text:bibliography-mark>
</text:p>
```



This paragraph[1], written on Thursday, July 11, 2002, shows how **easy** it is to extract plain text.
[ART00]¶
¶

Table Model Details

- CALS table model
  - Only **one** table model
  - Similar to HTML: table, rows, cells
  - Only cells contain content (or subtables)
  - Column + row spanning, covered cells
- Cell content
  - Text paragraphs
  - Attributes for formulas and values

Table Model, cont.

- Cells contain: formatting, formulas, values

```
<table:table-cell table:style-name="Tabelle1.A1"
                  table:value-type="string">
 <text:p text:style-name="Table Heading">Hello</text:p>
</table:table-cell>

<table:table-cell table:style-name="Table1.B2"
                  table:formula="<A2>*<A2>"
                  table:value-type="float"
                  table:value="1.234321">
 <text:p text:style-name="P2">01.23</text:p>
</table:table-cell>
```

Interoperability

- Exchange Formats
  - New StarOffice filters
    - WPS 2000, Ichitaro 10, HWP 97
    - Developed by external companies
    - Generates XML File Format as output from native apps
  - DocBook transformation
    - Developed by open community
    - Converts between DocBook an OpenOffice.org XML
    - Goal: OOo as DocBook editor
    - http://www.chez.com/ebellot/ooo2sdbk/ (French)