



# InteropGrabBag in LibreOffice Writer

Miklos Vajna

2014-02-01



# Filter problems

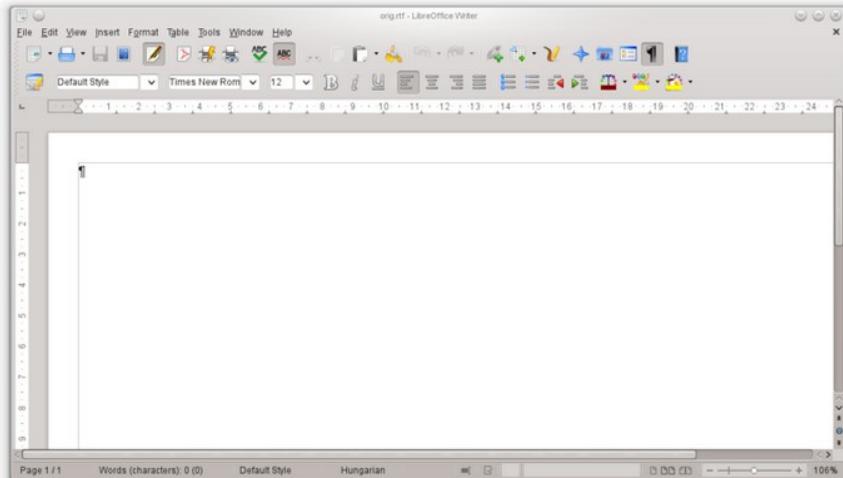


# What is a filter problem?

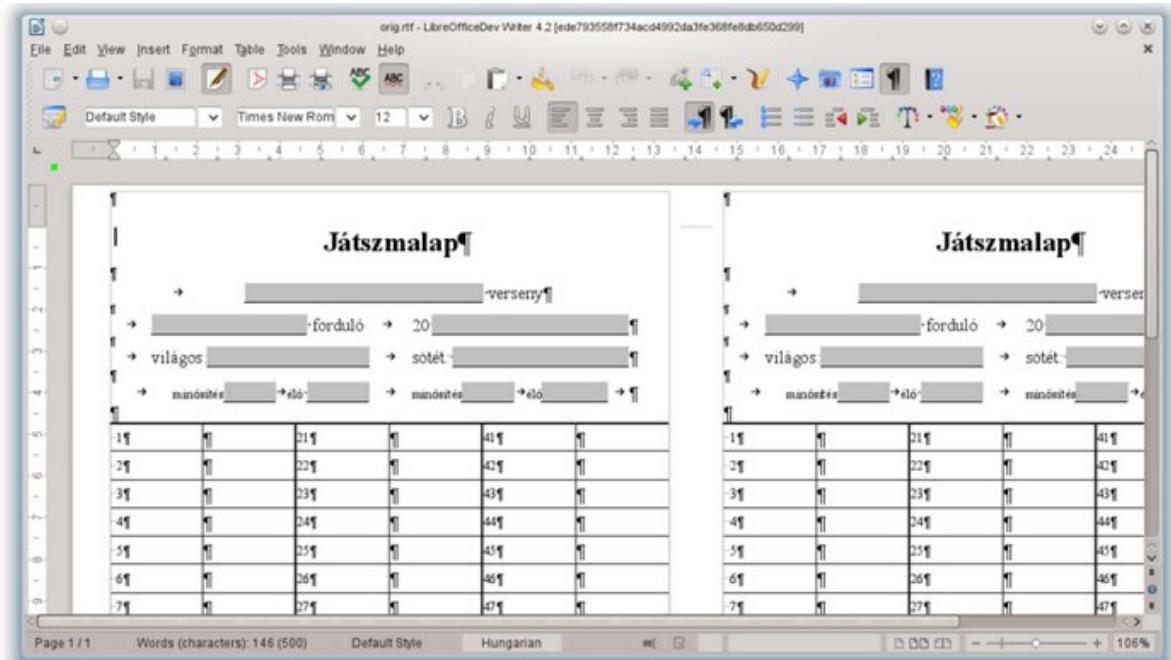
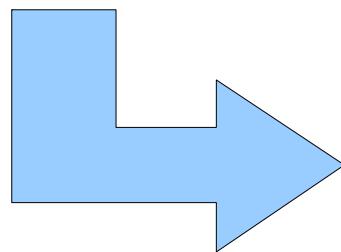
- Filters map between an input/output stream and the document model
- A given filter: given file format
- If a feature in a file format is not roundtripped perfectly through the document model



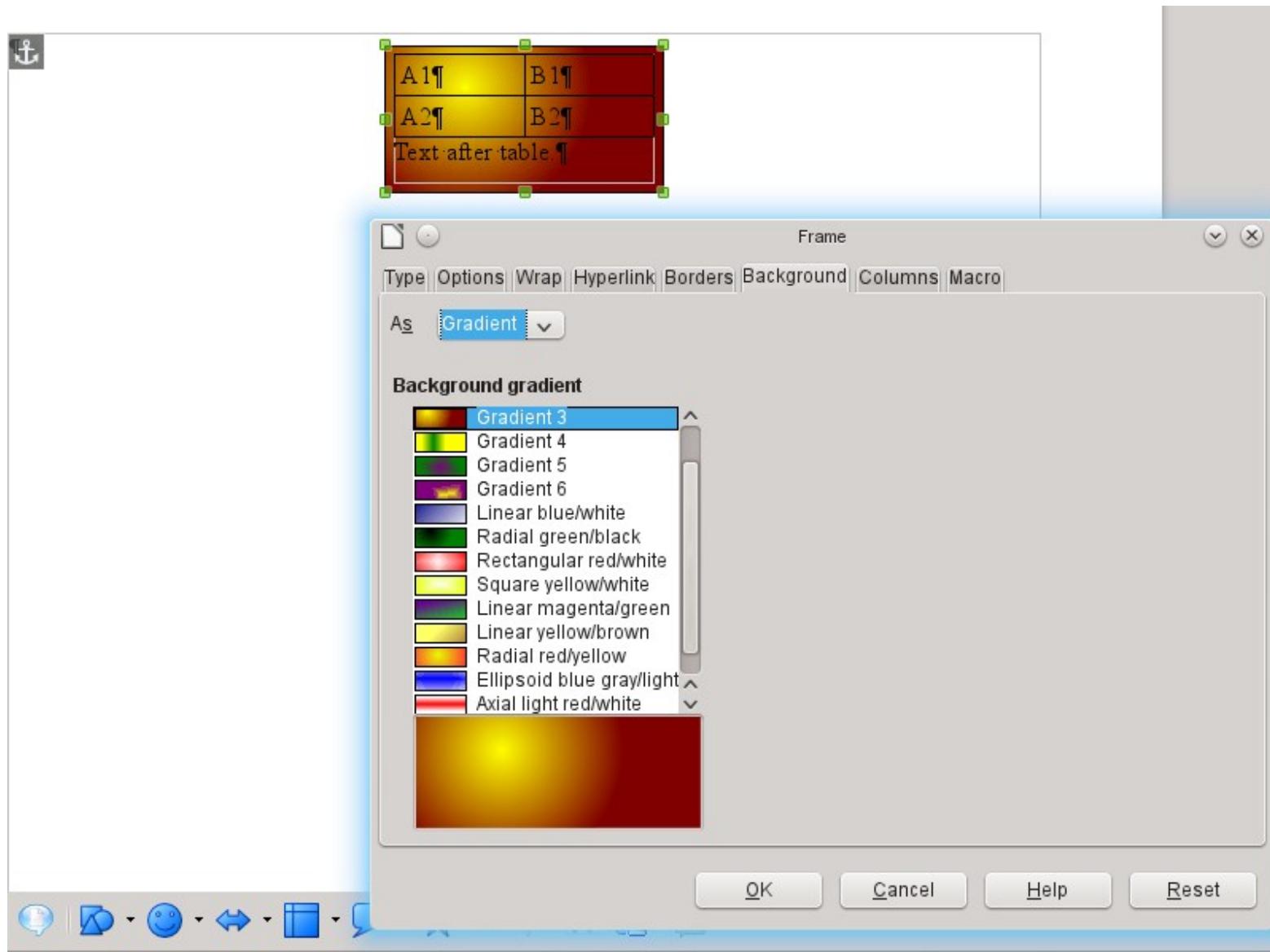
# Case #0: feature is already in core



- RTF import,  
groupshape support



# Case #1: Implement the feature in core



# Case #2: Import the result of the feature in the filter

- E.g. Table styles, document themes
- SmartArt import
- On import, apply the result of that as direct formatting
- Better, than nothing, but no real editing can be performed



# InteropGrabBag



# Case #3: Just preserve it

- If a feature is completely unsupported by core, it makes sense to first just preserve it
- Use case:
  - Long document
  - Full of complicated features
  - Just want to correct a typo



# We do it for ODF already

- `css::xml::ParaUserDefinedAttributesSupplier`
- `css::xml::TextUserDefinedAttributesSupplier`
- `css::xml::UserDefinedAttributesSupplier`
- “The idea behind this property is that a parser can throw away all attributes that it cannot handle by itself (...) can be written back without loss.”
- WW8 as well:  
`SwTOXBase::maMSTOCExpression`



# Why a new API?

- Need separate storage
  - ODF → foreign format
    - Would need manual mapping anyway
- Need a more flexible data structure
  - UserDefinedAttributes is just a string-string map
  - We want to store nested structures as well



# The InteropGrabBag API

- New SfxPoolItem: SfxGrabBagItem
- Writer: RES\_{CHR,PAR}ATR\_GRABBAG, etc.
- UNO: InteropGrabBag property
  - Type: css::beans::PropertyValues
    - Effectively a string-any map



# A set of bags

- Each UNO object may implement it, currently supported:
  - `css::document::OfficeDocument`
  - `css::drawing::Shape`
  - `css::style::CharacterProperties`
  - `css::style::ParagraphProperties`
  - `css::style::Style`
  - `css::text::BaseFrameProperties`



# The danger of hidden properties

- Problem: e.g. paragraph border is not supported, hidden “big black border” property
- User copy&pastes it elsewhere
- Result: pasted paragraph still has that property, probably not wanted
- Solution: clear InteropGrabBag on copying / when object is altered



# Just for Writer?

- The API is generic to be used in Calc, Impress, etc.
- Currently mostly implemented for Writer only
- Just for OOXML?
  - It's the only user ATM
  - Different properties can happy co-exist
    - Most of current keys are prefixed with OOX anyway



# Conclusions

- The end result: much improved OOXML roundtrip
- Now perfect roundtrip examples (incomplete list):
  - Document themes
  - SmartArt
  - Table styles

