Mail Merge Internals

Eilidh McAdam

Mail Merge

- Mail merge fills a template from a datasource, producing a single or separate documents
  - Datasource includes databases and spreadsheets
  - Also used to generate labels and envelopes
- Used by organizations – government, business, non-profit
  - Anyone that requires form letters
- Around since the 80s

Mail Merge Issues

- Overlooked due to userbase
- 106 issues in Bugzilla
  - Most are bugs but there are a few feature requests
- Over a decade old
- Hard to test

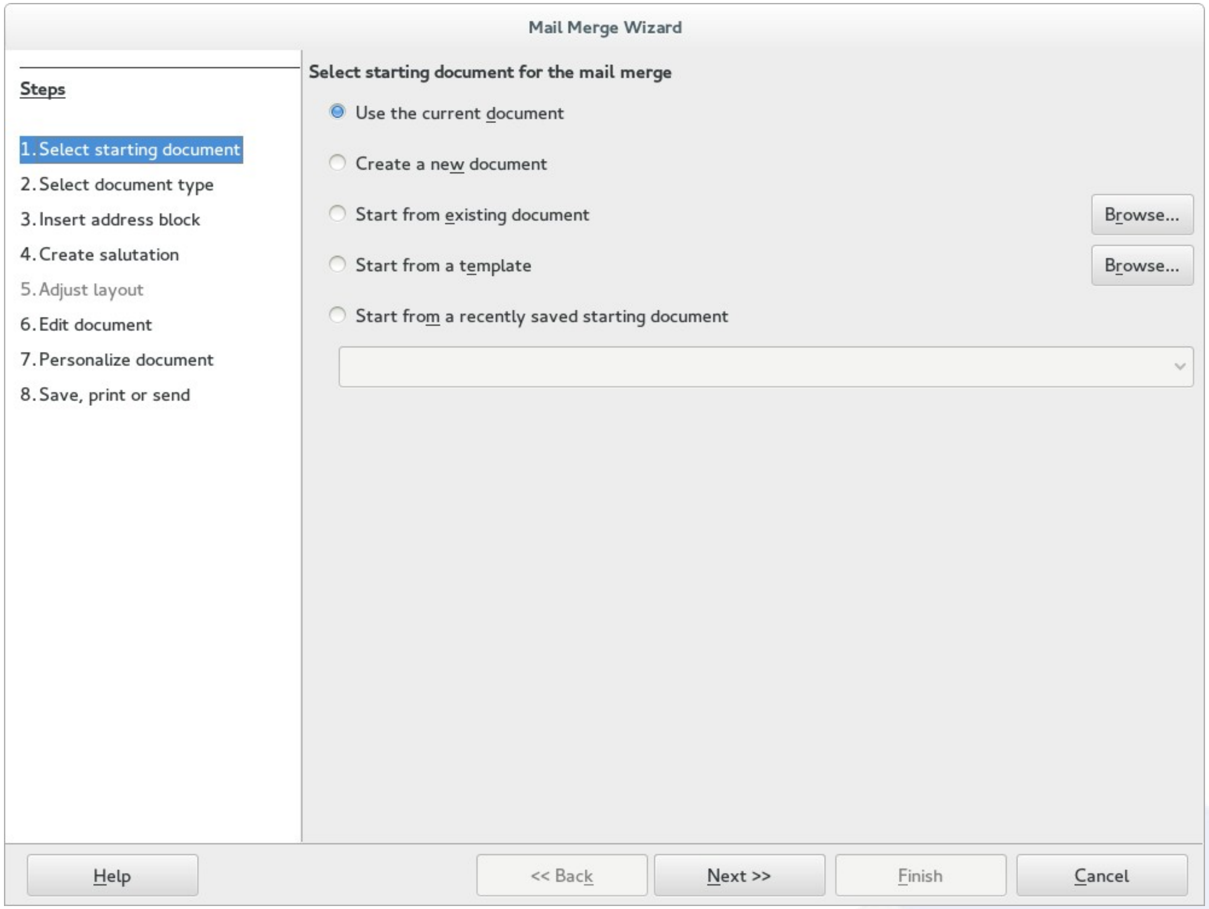
Performing Mail Merge

- Process:
  - Start with source document containing fields
  - Specify the type of merge and a datasource
  - Print, view or save the resulting documents
- Several points of entry:
  - Mail merge wizard
  - Print dialog
  - UNO
  - Labels and envelopes

Mail Merge Wizard

- Tools - Mail Merge Wizard...
- Uses roadmap format to obtain merge information
- Allows customisation of merge type, data source, address block and salutation
- Merge is performed implicitly (on page change)
- Final action is to print or save the resulting documents
- Allows editing of individual merged documents
- Code: sw/source/ui/dbui/
  - mailmergewizard.cxx, mm*page.*xx

Mail Merge Wizard



Mail Merge Wizard

Steps

1. Select starting document
2. Select document type
3. Insert address block
4. Create salutation
5. Adjust layout
6. Edit document
7. Personalize document
8. Save, print or send

Select starting document for the mail merge

● Use the current document

○ Create a new document

○ Start from existing document

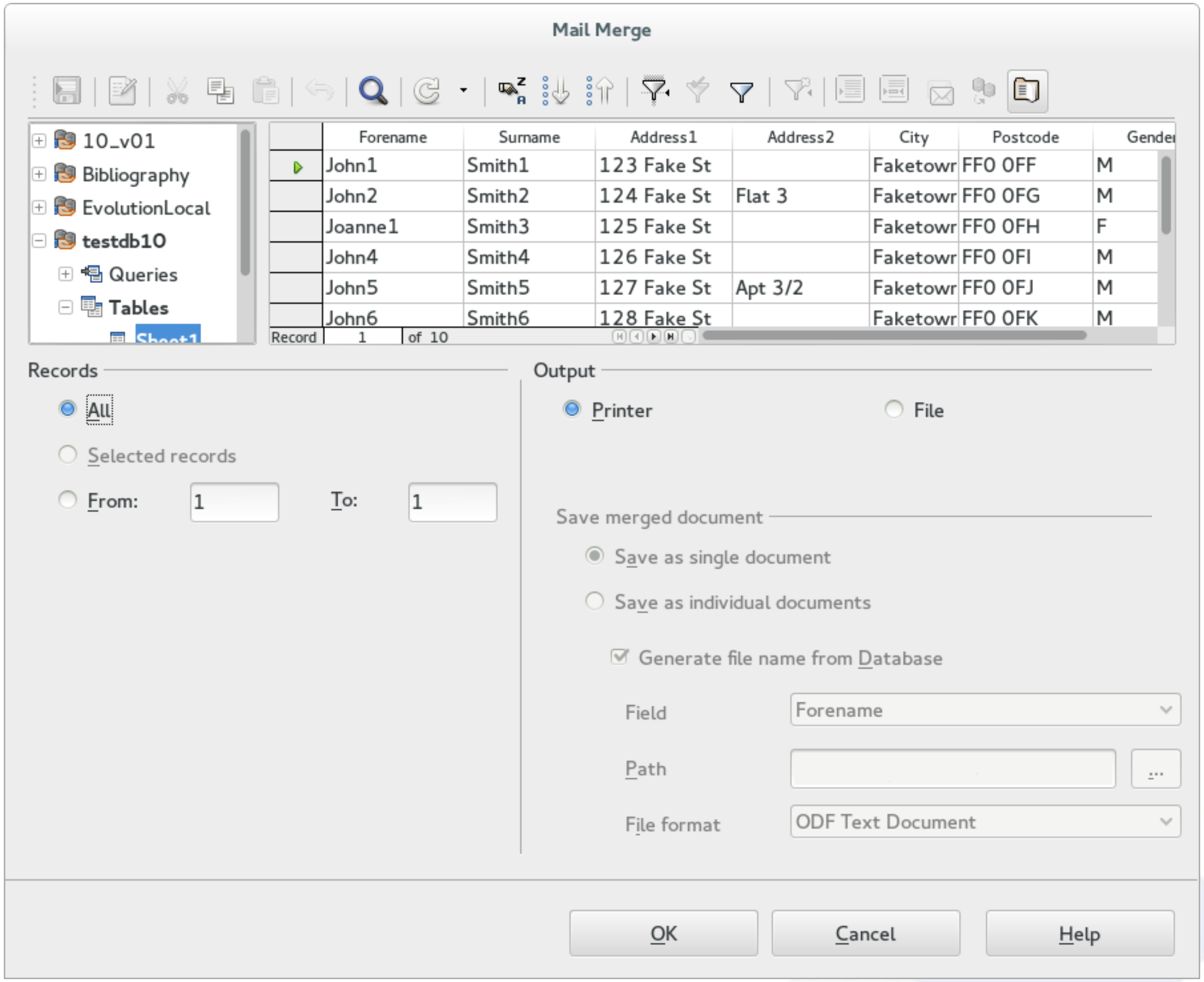
○ Start from a template

○ Start from a recently saved starting document

Merge via File - Print

- Given the option of using this dialog when printing with merge fields in a document
- Has subset of merge wizard functionality
- Allows more flexible data selection and filtering
- SwMailMergeDlg code:
  - sw/source/ui/envelp/mailmrge.*xx

Merge via File - Print



Mail Merge

- 10_v01
- Bibliography
- EvolutionLocal
- testdb10
  - Queries
  - Tables
    - Sheet1

	Forename	Surname	Address1	Address2	City	Postcode	Gende
▷	John1	Smith1	123 Fake St		Faketowr	FF0 0FF	M
	John2	Smith2	124 Fake St	Flat 3	Faketowr	FF0 0FG	M
	Joanne1	Smith3	125 Fake St		Faketowr	FF0 0FH	F
	John4	Smith4	126 Fake St		Faketowr	FF0 0FI	M
	John5	Smith5	127 Fake St	Apt 3/2	Faketowr	FF0 0FJ	M
	John6	Smith6	128 Fake St		Faketowr	FF0 0FK	M

Record 1 of 10

Records

● All

○ Selected records

○ From: 1 To: 1

Output

● Printer ○ File

Save merged document

● Save as single document

○ Save as individual documents

☑ Generate file name from Database

Field: Forename

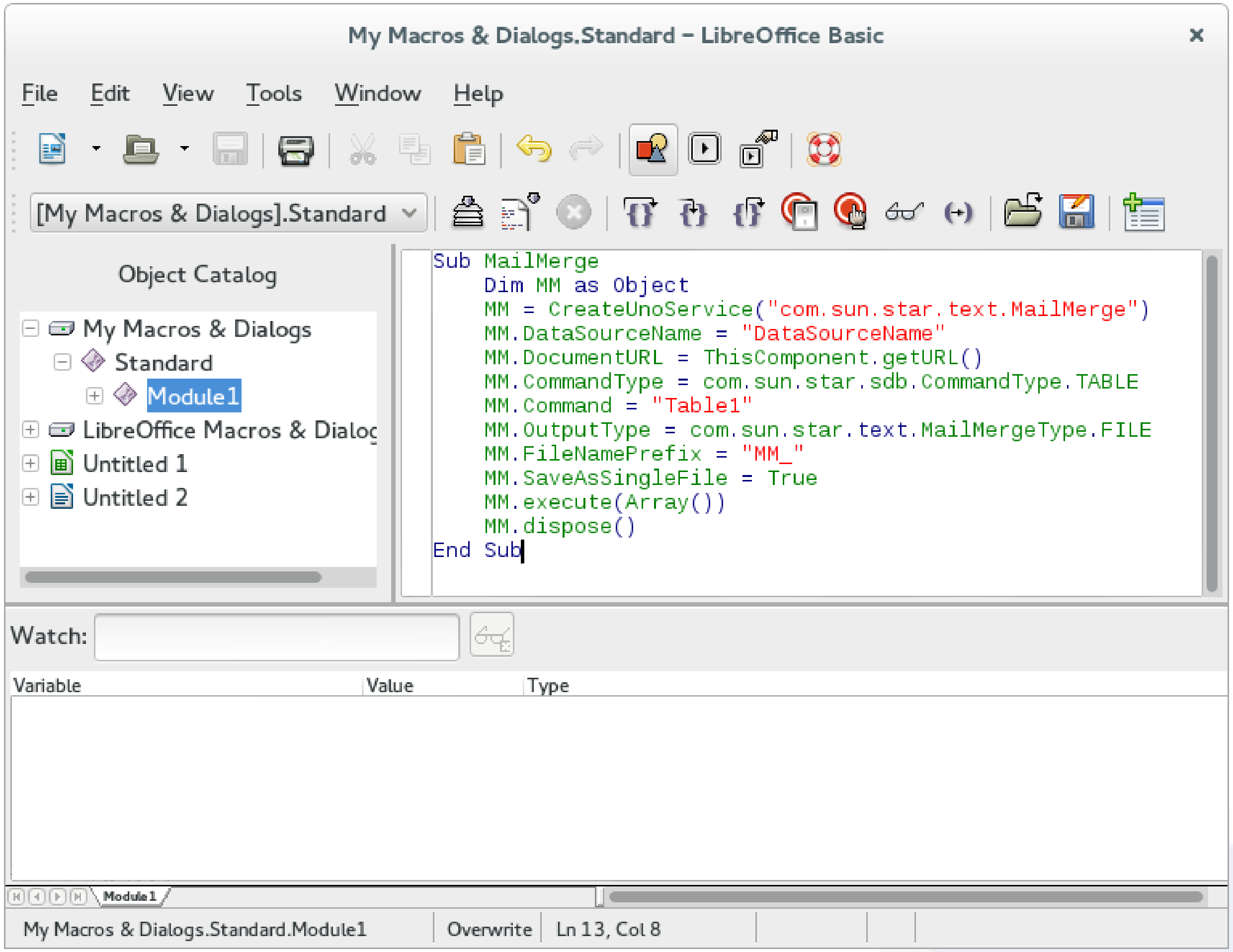
Path:

File format: ODF Text Document

Mail Merge via UNO

- Uses UNO MailMerge service
- Performs a silent merge
- All settings are specified up front
- sw/source/uibase/uno/unomailmerge.cxx
  - SwXMailMerge::execute

Mail Merge via UNO



My Macros & Dialogs.Standard – LibreOffice Basic
File Edit View Tools Window Help
[My Macros & Dialogs].Standard
Object Catalog
My Macros & Dialogs
Standard
Module1
LibreOffice Macros & Dialog
Untitled 1
Untitled 2
Watch:
Variable
Value
Type
Module1
My Macros & Dialogs.Standard.Module1
Overwrite
Ln 13, Col 8

```
Sub MailMerge
    Dim MM as Object
    MM = CreateUnoService("com.sun.star.text.MailMerge")
    MM.DataSourceName = "DataSourceName"
    MM.DocumentURL = ThisComponent.getURL()
    MM.CommandType = com.sun.star.sdb.CommandType.TABLE
    MM.Command = "Table1"
    MM.OutputType = com.sun.star.text.MailMergeType.FILE
    MM.FileNamePrefix = "MM_"
    MM.SaveAsSingleFile = True
    MM.execute(Array())
    MM.dispose()
End Sub
```

Mail Merge Code

- SwDBManager
  - Handles intersection of documents and data
  - Responsible for resolving fields from a data source
  - sw/source/uibase/dbui/dbmgr.cxx
- Contains two main mail merge functions
  - MergeDocuments(SwMailMergeConfigItem&, SwView&)
    - Called from mail merge wizard
  - MergeMailFiles(SwWrtShell*, const SwMergeDescriptor&)
    - Called from SwDBManager::MergeNew

Mail Merge Logic

- Merge dataset is set up
- A monitor dialog is initialised if merge is non-silent
- Three document shells involved:
  - Source document
    - Passed in as function argument
  - Target document
    - Created as hidden document by merge function
  - Working document
    - Created to contain merge result of current record

Mail Merge Logic

- Per record:
  - Source document is copied into working copy

    SwDoc::CreateCopy
  - Fields are updated and converted to text

    SwDoc::UpdateFlds and SwDoc::ConvertFieldsToText
  - Styles are renamed and copied

    For header and footer handling
  - Result is pasted into target document

    SwFEShell::Paste

MergeDocuments()

- sal_Int32 SwDBManager::MergeDocuments(

  SwMailMergeConfigItem& rMMConfig,

  SwView& rSourceView)
- Called via mail merge wizard
- Pastes all documents into single target document, regardless of desired output
  - Wizard code handles slicing, saving, printing and emailling
- Passes resulting target document back out via config item

MergeMailFiles()

- bool SwDBManager::MergeMailFiles(

  SwWrtShell* pSourceShell,

  const SwMergeDescriptor& rMergeDescriptor)
- Handles emailing, printing and saving
- Saves individual files as soon as they are constructed
- Over 500 lines long
- Many code paths dependent on booleans derived from parameters

Mail Merge Activity

- Collaboration with Munich
  - Issues stemming from other parts of the codebase
  - Lots of fixes from Jan Marek Glogowski
- Refactoring out MergeDocuments()
  - MergeMailFiles() is more general purpose
  - Eases maintenance and enhancements
- Still plenty of potential for improvement

Thank you

- Questions?
- Itomig information
  - eilidh.mcadam@itomig.de
  - IT service management
    - Consultancy, system integration, development
  - Free Office solutions (LibreOffice)
    - Migration, macro programing, extension development, application development