

# LibLibreOffice

## *Re-using LibreOffice ...*

Michael Meeks <[michael.meeks@collabora.com](mailto:michael.meeks@collabora.com)>

mmEEKS, #libreoffice-dev, irc.freenode.net

*“Stand at the crossroads and look; ask for the  
ancient paths, ask where the good way is,  
and walk in it, and you will find rest for your  
souls...” - Jeremiah 6:16*



# Re-using LibreOffice

- talk: to unoconv
  - Re-use a factory process
  - use BinaryURP
  - suffer variously ...
- fork: soffice –convert-to
  - Trivial: spawn and chew the output
- link: liblibreoffice
  - direct C / C++ integration / control.
  - various optimisation opportunities



# Command-line soffice

```
./soffice -help ...
```

```
--convert-to output_file_extension[:output_filter_name] [--outdir  
output_dir] files
```

Batch convert files.

If --outdir is not specified then current working dir is used as output\_dir.

Eg. --convert-to pdf \*.doc

```
--convert-to pdf:writer_pdf_Export --outdir /home/user  
*.doc
```

```
--print-to-file [-printer-name printer_name] [--outdir  
output_dir] files
```

Batch print files to file.

If --outdir is not specified then current working dir is used as output\_dir.

Eg. --print-to-file \*.doc

```
--print-to-file --printer-name nasty_lowres_printer  
--outdir /home/user *.doc
```



# Command-line issues

- spawning does setup every time
  - Parsing tens of thousands of lines of config etc. endless bootstrapping ...
- Error handling is not ideal
  - Did the file turn up, did the process crash ? Timeouts / etc.
- On the other hand v. easy, scriptable etc.



# LibLibreOffice ...

- A very simple ~header-only API
  - fully abstract: fn pointers, opaque structs etc.
  - To be fair you need to dlopen / do one dlsym:
    - liblibreoffice\_hook ...
  - Native C API
    - C++ wrappers around that ...
  - No sockets opened, no plugins / simple init.
  - Global error messages.
- Lots more work to do etc.



# Hooking a symbol ...

```
typedef LibreOffice *(HookFunction)(void);  
SAL_DLLPUBLIC_EXPORT LibreOffice *lo_init( const char *install_path )  
{  
    char *imp_lib;  
    void *dlhandle;  
    HookFunction *pSym;  
    imp_lib = (char *) malloc( strlen (install_path) +  
                               sizeof( TARGET_LIB ) + 2 );  
    strcpy( imp_lib, install_path );  
    strcat( imp_lib, "/" );  
    strcat( imp_lib, TARGET_LIB );  
    dlhandle = dlopen( imp_lib, RTLD_LAZY ) )  
    pSym = (HookFunction *) dlsym( dlhandle, "liblibreoffice_hook" );  
    free( imp_lib );  
    return pSym();  
}
```

Pass the path: /usr/lib/libreoffice/

[ All error handling removed  
for clarity ]



# Skeletal C API

```
struct _LibreOffice {  
    int nSize;  
  
    void (*destroy) (LibreOffice *pThis);  
  
    int (*initialize) (LibreOffice *pThis,  
                      const char *pInstallPath);  
  
    LibreOfficeDocument *(*documentLoad) (LibreOffice *pThis,  
                                           const char *pURL);  
  
    char * (*getError) (LibreOffice *pThis);  
};
```

```
struct _LibreOfficeDocument {  
    int nSize;  
  
    void (*destroy) (LibreOfficeDocument *pThis);  
  
    int (*saveAs) (LibreOfficeDocument *pThis,  
                  const char *pUrl, const char *pFormat);  
};
```

***Plus an inline C++ API wrapper  
that evaporates away at compile  
time.***



# Other bits ...

- Shrunk Libraries ... new work here from Matus / Stephan / Kendy / Tor.
- Tiled rendering output ...

```
VirtualDevice aDevice;  
pViewShell->PaintTile(aDevice, contextWidth, contextHeight,  
                      tilePosX, tilePosY, tileWidth, tileHeight);  
Bitmap aBitmap(aDevice.GetBitmap(aDevice.PixelToLogic(Point(1,0)),  
                                aDevice.PixelToLogic(Size(contextWidth, contextHeight))));
```

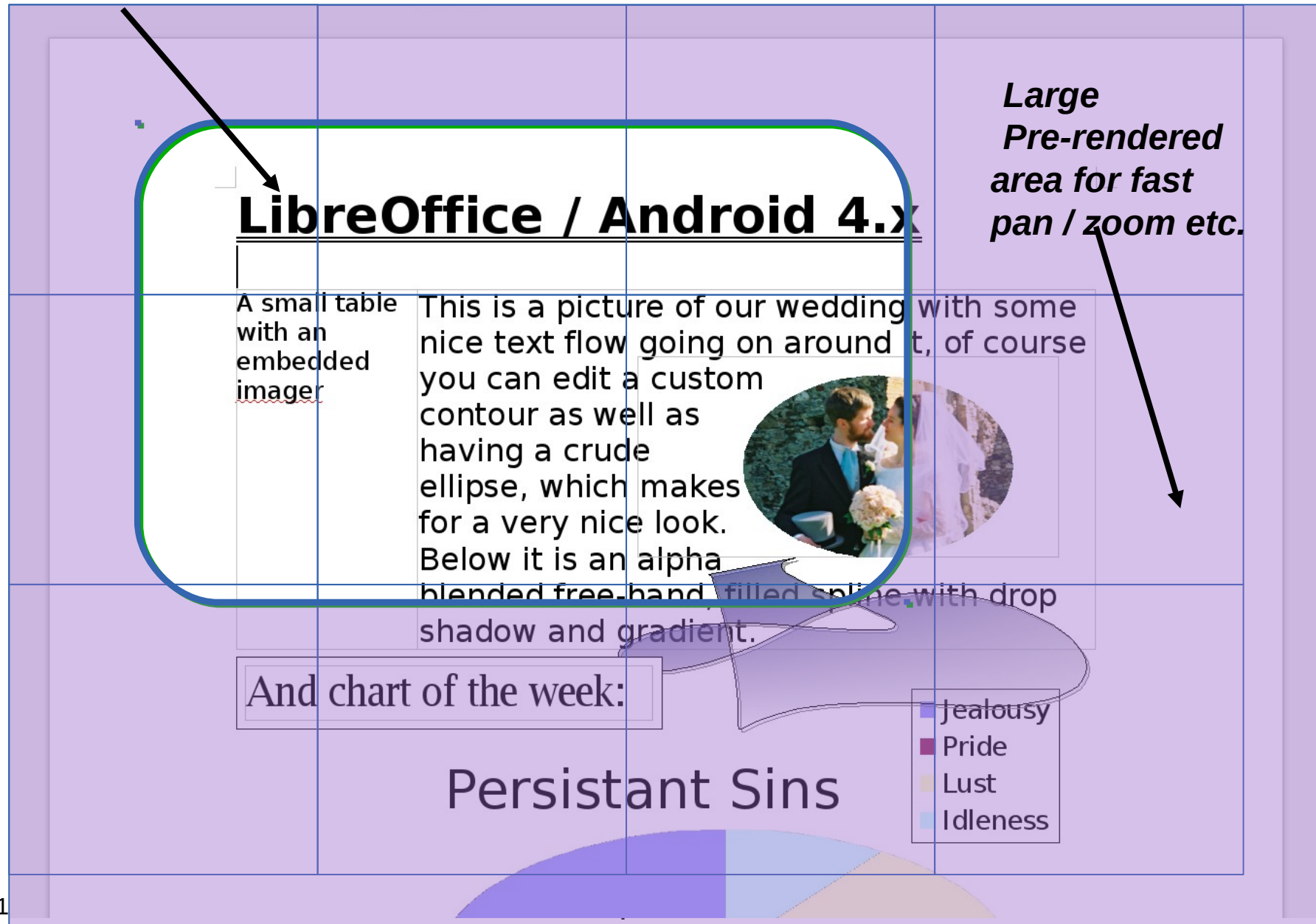
- Future work:
  - A 'kdeinit' solution: Init then fork
    - Avoids XML re-parsing etc.
  - Flush-caches ...
    - Avoid leaks with global cache flushes
    - GraphicObjectManager awfulness ...





# Fast Tiled Rendering to bitmaps

*Tablet display*



# Demo app in instdir/

- Simple tiled rendering test app in  
sw/qa/tiledrendering/ - 200x lines ...
  - Still requires VCL & hard linking; needs  
abstracting & hiding etc.





# LibLibreOffice Conclusion

- **Lots of ways to re-use LibreOffice**
  - Liblibreoffice: new, and interesting approach.
    - Leaves you fully in control of your process management.
  - Ideal for GNOME-Documents (eg.)
  - More fun stuff to come.
  - Hope.
- **Thanks for all of your help and support !**

*Oh, that my words were recorded, that they were written on a scroll, that they were inscribed with an iron tool on lead, or engraved in rock for ever! I know that my Redeemer lives, and that in the end he will stand upon the earth. And though this body has been destroyed yet in my flesh I will see God, I myself will see him, with my own eyes - I and not another. How my heart yearns within me. - Job 19: 23-27*

