



Overview

1. Some MacOS X Basics
2. VisualWorks and Cocoa
3. Standalone Applications written
in VisualWorks
4. Current Status of the Port



Some MacOS X Basics



MacOS X is a new Operating System based on Unix

The biggest difference to other Unixes:

It does not use X-Windows for display operations

From the UNIX perspective:

- its based on a Mach Kernel, provides preemptive multitasking, POSIX Threads, BSD Sockets, and many other unix style facilities
- there is a Terminal and a command prompt

For the Macintosh User, its important that:

- the window server is more complex than X11
- the file system supports type/creator settings
- migrating from MacOS 9 to MacOS X is easy because old applications still run in the Classic environment
- you don't need the Terminal and the command prompt



MacOS X for Developers



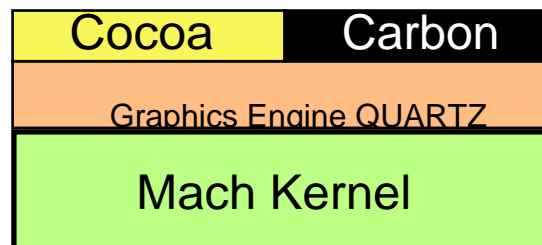
Two development platforms are supported:

Carbon:

- C-library of functions already established on MacOS 9,
- porting of old apps is much easier,

Cocoa:

- completely new class library, derived from NextStep
- Objective-C based





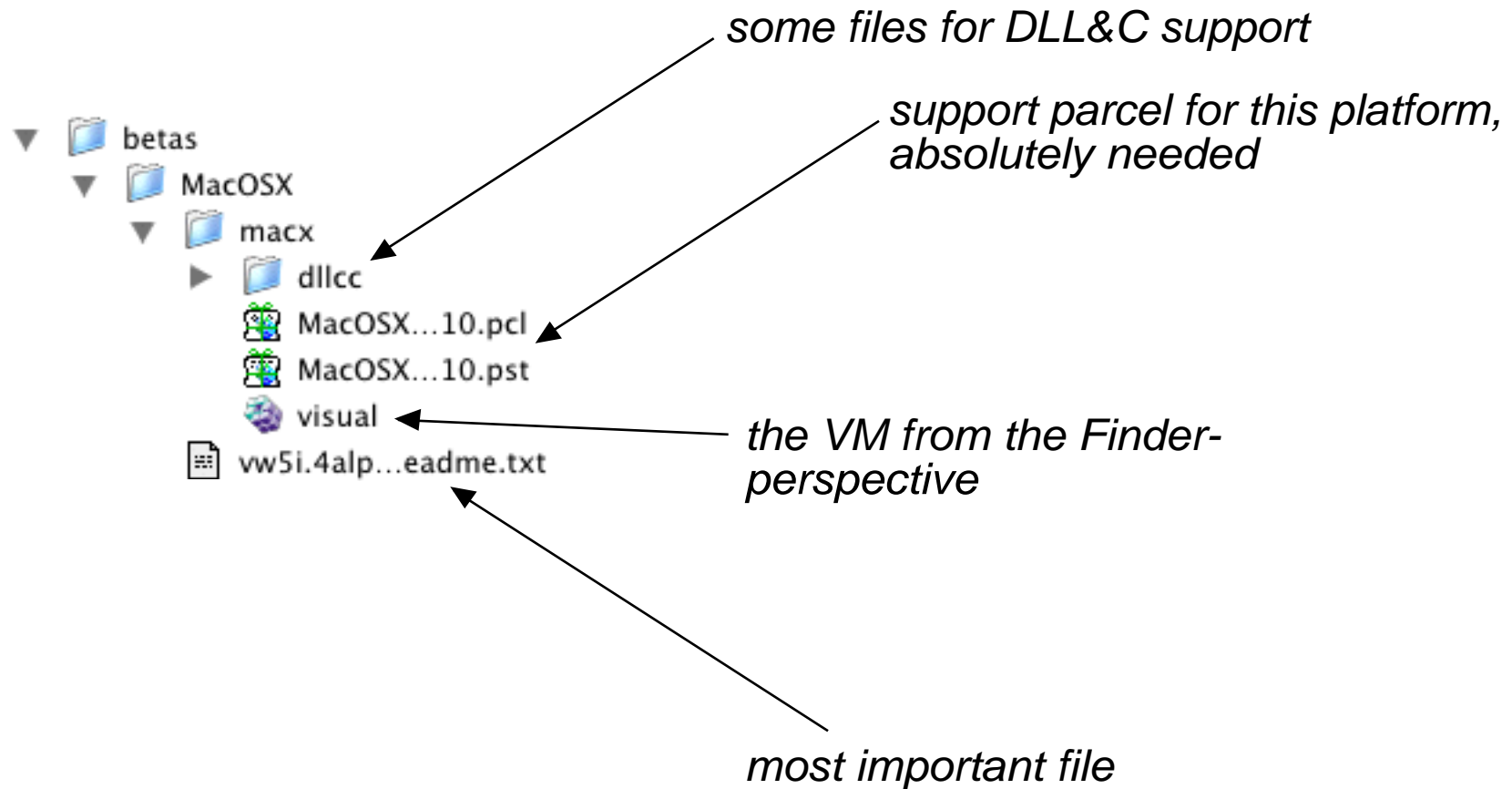
VisualWorks and Cocoa



- The VisualWorks port to MacOS X is a Cocoa application
- The base machinery is a standard UNIX VM, written in C
- The display part is totally new, and uses Objective-C objects to interface to the window server
- The executable can be used from a Terminal command line with no resources
- As a real MacOS X application, it appears on the desktop and in the Finder and has proper resources



The VisualWorks Application





The Unix Perspective



visual.app/

- this name only visible from a Shell

Contents/

PkgInfo
Info.plist

- file containing type/creator bits as in MacOS 9
- XML file containing all package infos of this application

MacOS/

visual

- the Unix type executable

Resources/

MainIcons.icns
ImageIcons.icns
herald.tiff
resource.im

- the icons for the application
- the icons for image files
- the herald screen graphics
- **OPTIONAL:**
place and name for an image bundled with a VM

English.lproj/

InfoPlist.strings
<other files>

- place for language dependent resources for menus etc.
- Unicode encode file with language dependent strings



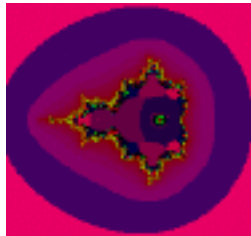
The Mac Perspective



- As a real MacOS X application, the VM appears on the desktop and has proper resources
- A default image can be added to its resource fork
- By editing some application structure documents, you can personalize the application with startup picture, desktop icon, document icons, type/creator IDs



A Standalone Application



FractalExplorer.app/

- this is a deep copy of visual.app

Contents/

PkgInfo
Info.plist

- new type/creator

- adapted for new icons

MacOS/

visual

- the Unix type executable remains

Resources/

Fractal.icns
herald.tiff
resource.im

- new .icns file

- the herald screen graphics

- image prepared with Runtimepackager

English.lproj/

InfoPlist.strings
<other files>

- edit for new application name



Current Status of the Port



Because of the unusual window server interface, performance is suffering severely in the area of text and graphics display

PICs and THAPI has not been finished,
some primitives are not implemented

BSD Signals are not recognized properly, thus Sockets are not working correctly (especially UDP)

PLEASE: Report any comments and bugs to:
roland@heeg.de
ralf@heeg.de