

Welcome

To Advance through Presentation  
Use Page Up and Page Down Keys



99 | Worldwide  
Developers  
Conference



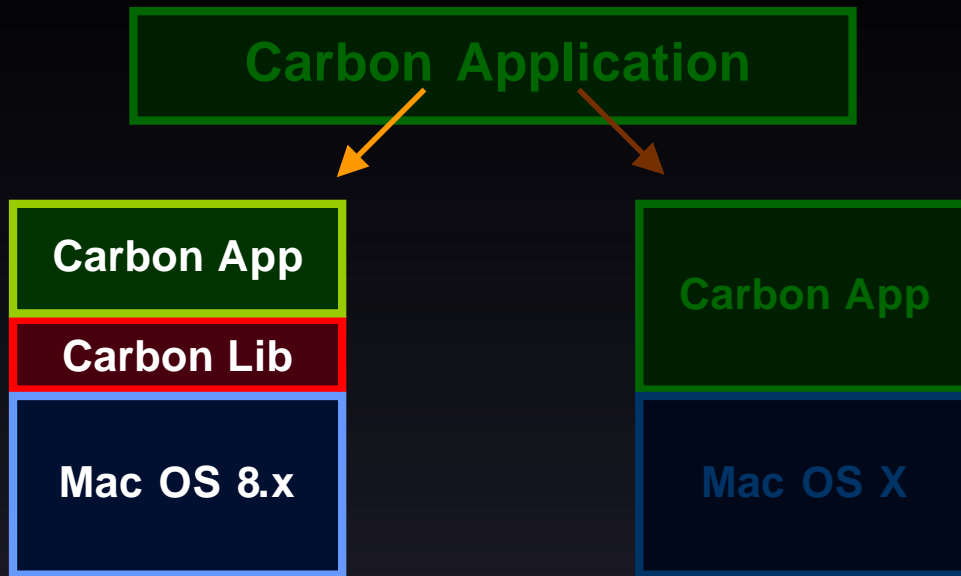
99 | Worldwide  
Developers  
Conference

# Carbon on Mac OS 8

Nitin Ganatra  
Carbon Team

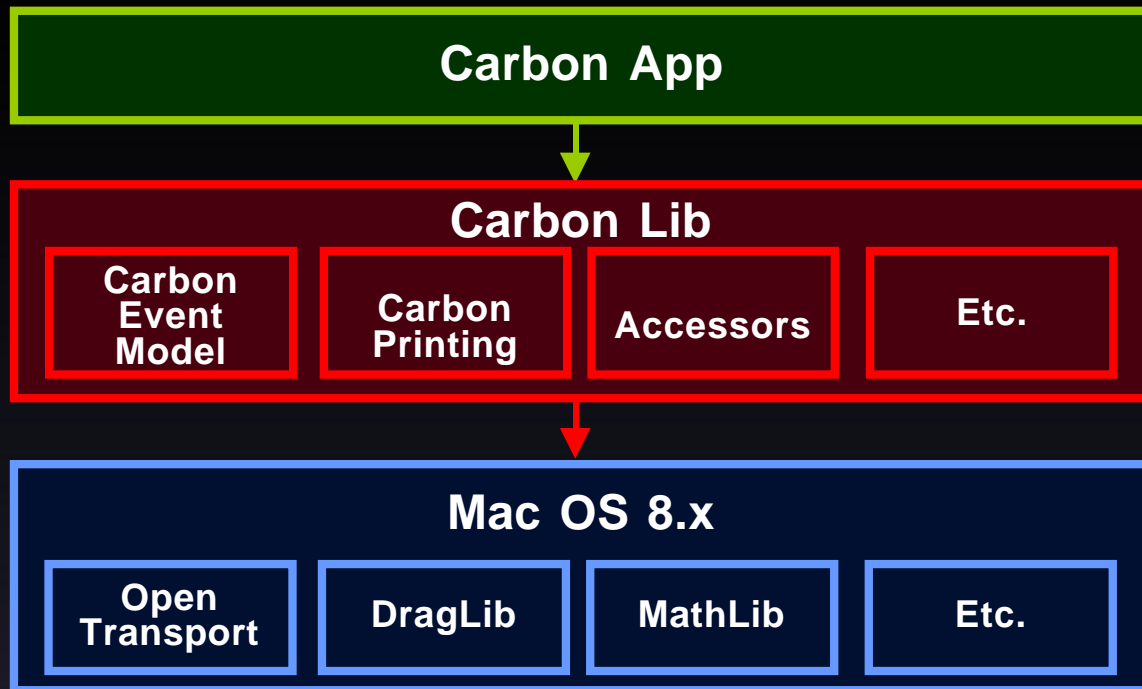
# What Is CarbonLib?

- A 'thin shim' that sits on top of 8.x



# CarbonLib and 8.x

- CarbonLib gives you everything



# Who Should Port?

- Applications (background and foreground)
- Plug-ins for Carbon applications
- No device drivers, cdevs or System extensions (INITs)



# Carbon Porting Strategy

- 8.x Development
- Carbon on 8.x SDK Development
- Carbon on OS X Development



# Before the SDK

- Port to PowerPC
- Latest Universal Interfaces



# Carbon SDK Development

- Carbon Universal Interfaces
- Compile with **TARGET\_CARBON=1**
- Likely Compiler errors...
- Link with CarbonLib...





# Likely Compiler Errors

- Opaque toolbox data structures
- GrafPtrs, WindowPtrs, DialogPtrs
- UniversalProcPtrs
- Misc.



# Opaque Structures

- WindowRecords, DialogRecords
- Menus
- Controls
- GrafPorts, Regions, QDGlobals
- AEDesc dataHandle



# Accessors

- Why are they opaque?
  - Easier bug fixes and feature enhancements
  - Thread-safety in the future
- Accessors don't allocate, don't deallocate.
- Use static globals to minimize allocations



# Simple Example

- Stack based storage

```
leftPosition = (**someRegion).rgnBBox.left;  
...
```

- Becomes

```
Rect regionRect;  
  
GetRegionBounds(someRegion, &regionRect);  
  
leftPosition = regionRect.left;  
...
```



# Another Example

- Getting a port's visible region:

```
static RgnHandle visRgn = NULL;
```

```
if (visRgn == NULL)  
    visRgn = NewRgn();
```

```
GetPortVisibleRegion(somePort, visRgn);
```

```
...
```



# GrafPorts and Friends

- On 8.x, a Dialog is a Window is a GrafPort
- On OS X, they're not
- Use 'casting functions' under Carbon

**GetWindowPort**  
**GetDialogWindow**

Etc.



# UniversalProcPtrs

- Mach-o is the native runtime on Mac OS X
- Still needed for runtime (CFM to Mach-o) interoperability
- **NewRoutineDescriptor** and **DisposeRoutineDescriptor** are gone
  - Use typed UPP calls instead



# UPP Example

```
ControlActionUPP someUPP =  
    NewRoutineDescriptor((ProcPtr) MyActionProc,  
        uppControlActionProcInfo, kPowerPCISA);
```

```
part = TrackControl(control, startPt, someUPP);  
DisposeRoutineDescriptor(someUPP);
```

- Becomes

```
ControlActionUPP someUPP =  
    NewControlActionUPP(MyActionProc);
```

```
part = TrackControl(control, startPt, someUPP);  
DisposeControlActionUPP(someUPP);
```





# Misc. Changes

- DefProcs repackaged
- OpenTransport API changes
- Thread Manager routines UPPified



# Link with CarbonLib

- Link only against CarbonLib
  - Linking to InterfaceLib and others will cause problems, to say the least
- Unsupported routines will generate linker errors...
  - The headers aren't done yet



# Libraries Replaced by CarbonLib

- InterfaceLib
- DragLib
- MathLib
- AppearanceLib
- QuicktimeLib
- And so many more...



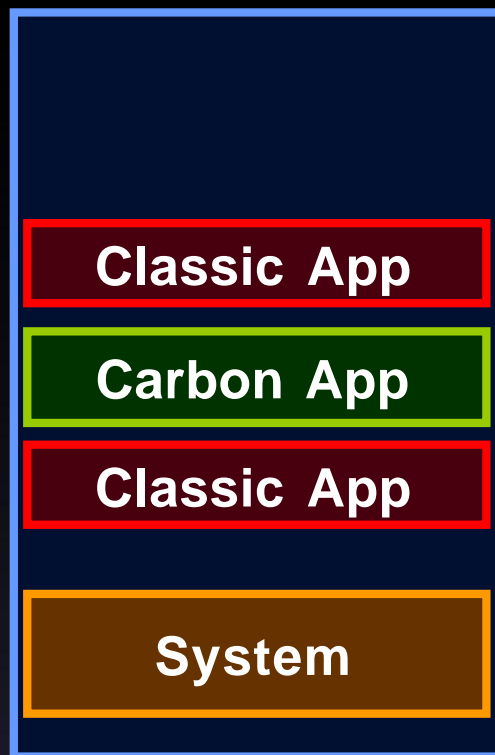
# Running Under CarbonLib on 8.x

- Not preemptively scheduled on 8.x
- Toolbox initialization
- Memory Management
- Memory Model
- Application plug-ins

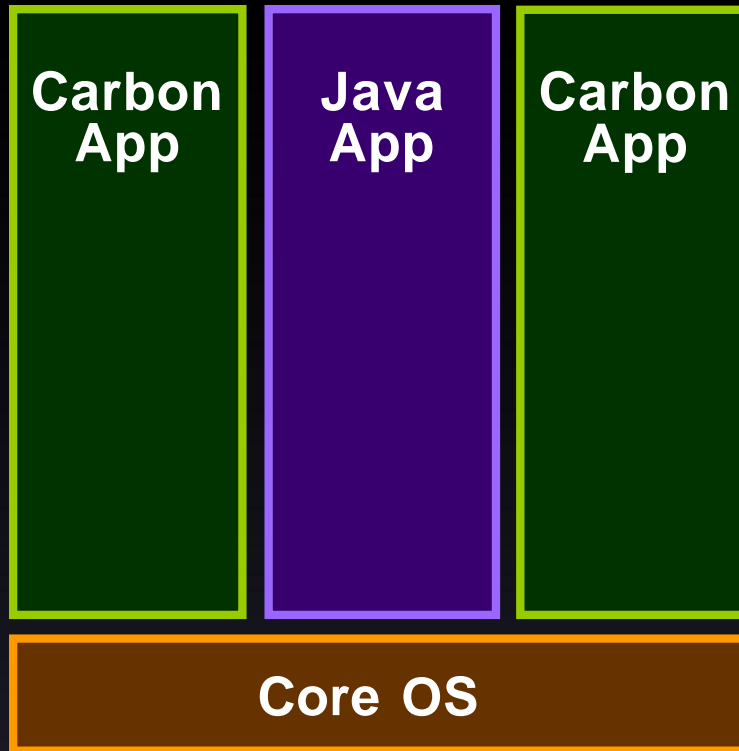


# 8.x Memory Model

- Look familiar?



# OS X Memory Model



# Memory Management

- Test at runtime for sparse VM  
**gestaltMemoryMapSparse**
- No more Zones
  - Assumed allocation behavior of underlying system.



# Plug-ins and Carbon

- Must be Carbonized just like apps
- Resource tracking and reclamation is up to the app
- CFPlugin and CFBundle can be used to help with plug-in management





# API Availability

- Older Systems have a subset of Carbon
  - Mac OS is moving too fast...
- Assume 8.1 baseline to run everywhere
- Additional Carbon API provided everywhere
- Everything else, test for it



# Available Everywhere

- Everything that's in 8.1
- Toolbox Accessor functions
- Navigation Services
- Control, Window, Menu Properties
- CoreFoundation
- Carbon Events
- The list goes on...



# Schedule

- CarbonLib will first ship integrated with Sonata system (fall '99)
- Standalone CarbonLib will ship a few weeks later



# Summary

- Only applications and their plug-ins should port to Carbon
- There are runtime differences, but most application behavior just works
- You still need to test for newer functionality



# So You Want To Start Porting?

- Switch to new Universal Interfaces
- Flip the **TARGET\_CARBON** compiler switch
- Link only against CarbonLib
- Carbon versions of PowerPlant and MacApp available
- Ship it!



# Related Sessions

---

## **Carbon:**

Carbon on Mac OS X

Hall A1  
**Tues., 4:00pm**

---

## **HLTB:**

Carbon Changes/Additions

Hall 2  
**Fri., 10:15am**

---

## **Mac OS X:**

Networking Overview

Hall A2  
**Fri., 9:00am**

---

## **Core Foundation:**

Plug-ins

Hall A1  
**Fri., 4:00pm**

---





99 | Worldwide  
Developers  
Conference

Q&A



Think different.<sup>TM</sup>





Welcome

To Advance through Presentation  
Use Page Up and Page Down Keys



99 | Worldwide  
Developers  
Conference