Welcome

To Advance through Presentation
Use Page Up and Page Down Keys

AppleScript, Mac OS X, and Carbon

Jason Yeo

AppleScript
Technology Manager

AppleScript, Mac OS X, and Carbon

Chris Espinosa

Manager, Components and Scripting

What's in This Session

- Overview of upcoming AppleScript features
- Writing Carbonized OSA Client applications
- Writing scriptable Carbon applications
- Writing scriptable Cocoa applications

Upcoming Features

- **using terms from** Block
- Apple events over IP
- Optional **on error**
- **file busy** property in **info for** result
- **OSAFlushTerminology** API

Demo

Upcoming Bug Fixes

- Large **typeMagnitudes** don't display correctly
- **run** (system event)
- Unicode text coercion memory problems
- Too-long NBP names crash
- Low-memory global accesses
- Relaunching delays

Upcoming Scriptability

- Keychain
- Sherlock II
- LaserWriter printer driver—**with properties** parameter for **print** event

Structural Changes

- Integrated English dialect
- All code is PowerPC in shared libraries—no accelerated code resources
- AppleScript is not Carbonized for Sonata, but can be used from Carbon clients

Carbonized OSA Apps

- Update to latest Interfaces and Libraries
- Set **TARGET_CARBON = 1**
- Fix the compile errors
- Remove all Mac OS link libraries and link with CarbonLib instead
- Compile and debug on Mac OS 8.6 with CarbonLib, then on Sonata
- Compile and run on Mac OS X

Main Carbon Issues

- No System heap in Mac OS X
  - All handlers and coercions are treated as application handlers
- Callback **UPPs**
  - **SendProc**
  - **ActiveProc**
- **dataHandle** field of **AEDesc** descriptors

Opaque AEDescs

- We're eliminating reliance on handle-based structures
- Like other toolbox structures, the **dataHandle** will be opaque, not necessarily a handle
- Accessors let you get at the contents of the handle
- In the future the data may not be there!

Using Accessors

```
#if !OPAQUE_TOOLBOX_STRUCTS
    count = GetHandleSize(desc->dataHandle);
    data = (char *)((unsigned char*)*desc->dataHandle);
#else
    count = AEGetDescDataSize(desc);
    data = NewPtr(count);
    AEGetDescData(desc, desc->descriptorType, data,
    count);
#endif
...
#if OPAQUE_TOOLBOX_STRUCTS
    DisposePtr(data)
#endif
```

Five Scenarios

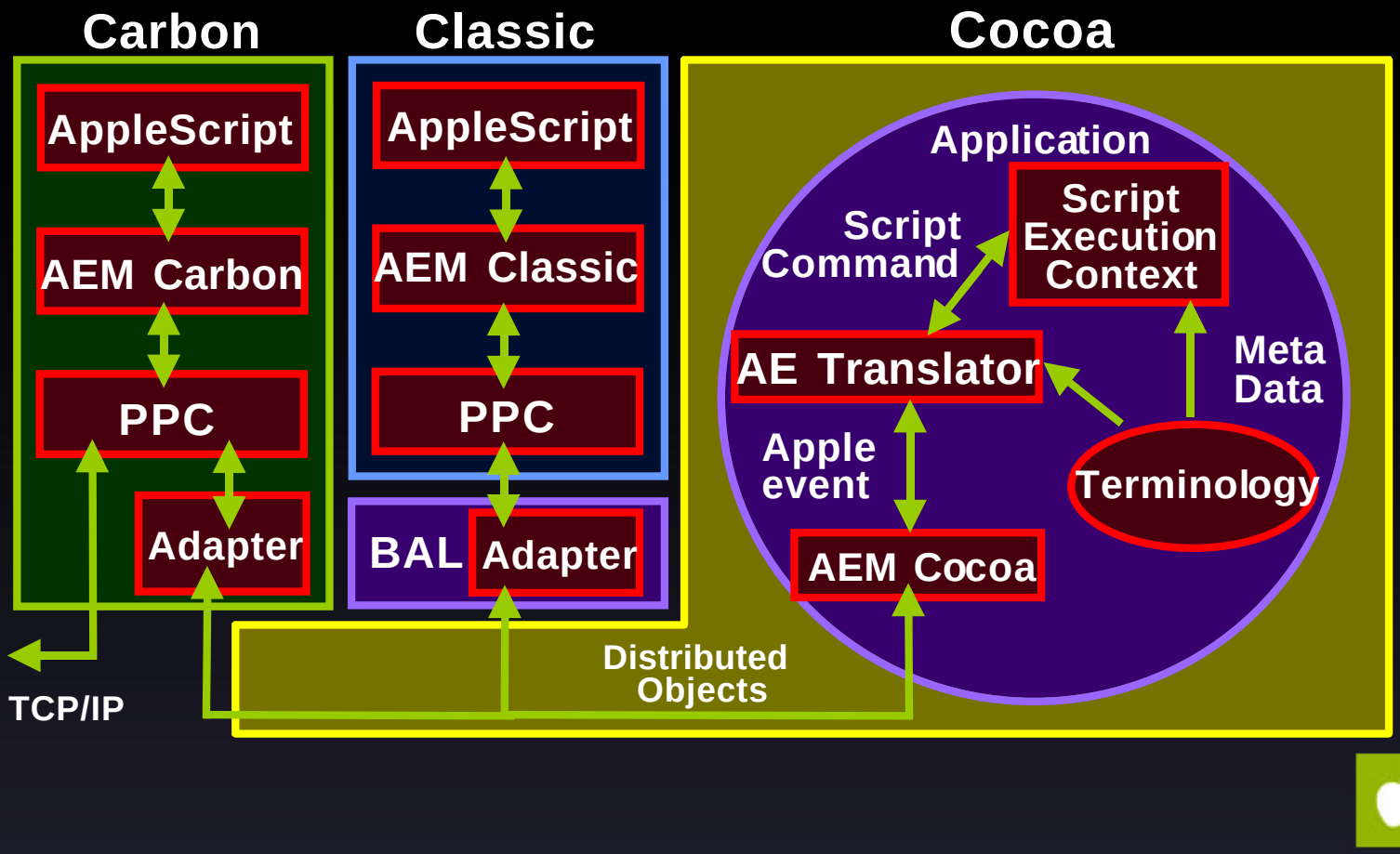
- Copy contents into struct
- Copy contents into variable-length pointer or handle
- Use contents as parameter to another routine (**const** or variable)
- Modify contents in place
- Assign contents by direct assignment

Examples

Carbonized OSAXen

- Won't run on Mac OS 8!
- Must be native Shared Libraries (see Extending AppleScript session)
- Set **TARGET_API_MAC_CARBON** and **TARGET_API_MAC_OSX**
- Carbonize as for an application
- Output .qtx and .qtr files

Scriptable Cocoa Apps



Summary

- New features upcoming in AppleScript: **using terms from** block, IP addressing
- You can write scriptable applications and OSA clients that run on both Sonata and Mac OS X
- Carbonization is straightforward but concentrate on opaque AEDescs

Roadmap

The Apple Event Model and Your Application To get started with AppleScript	Hall C **Thur., 9:00am**
Extending AppleScript For the new Standard Additions mechanism and Folder Actions	Hall B **Thur., 1:00pm**
AppleScript Feedback Forum For interactive feedback and discussion with the team	Hall C **Thur., 4:00pm**
AppleScript Birds of a Feather For community-building with other developers	Hall C **Thur., 5:30pm**

AppleScript Kitchens

London, U.K.:
June 15–17, 1999

Cupertino, CA:
August 17–19, 1999

For more information:
Email Jason Yeo at jason@apple.com

Q&A

Think different™

Welcome

To Advance through Presentation
Use Page Up and Page Down Keys