

Welcome

To Advance through Presentation
Use Page Up and Page Down Keys



99 | Worldwide
Developers
Conference



AppleScript Birds of a Feather

Cal Simone

AppleScript Terminologist

Problems Facing Scripters

- Inconsistent terminology
- Poor choice of terminology
- Poor object model support
- No object model support



Problems Facing Developers

- Lack of style guidelines
- Lack of decent, applicable sample code
- Inconsistent/weird support in frameworks
- No ratification or certification
- No way for users to determine good implementations










The Scriptability Scorecard

- Inclusion of specific features
- Overall qualities of scripting implementation



The Scriptability Scorecard

Scale of 0–100

| | |
|-------------------|---|
| 1–10: ½ script |  |
| 11–20: 1 script |  |
| ... | |
| 91–100: 5 scripts |      |



Requirements for Rating

- Have terminology resource (‘aete’)
- Go beyond required events and Do Script
- “Implement a meaningful and reasonably useful set of verbs and classes”



Phase 1:

Scripting Features Supported

Levels of Scriptability

- Is the feature supported?
- Conformity to standards does count, but otherwise doesn't rate the quality



Phase 1:

Scripting Features Supported (40 points)

Levels of Scriptability

- Object model support
- Recordable
- Attachable
- Embeddable



Phase 1:

Scripting Features

Object Model Support



Phase 1:

Scripting Features

Object Model Support

- Promote consistency and intuitiveness

Get the color of the first picture

not

GetColor



Phase 1:

Scripting Features

Object Model Support

- Promote consistency and intuitiveness

Get the color of the first picture

not

GetColor

- Lots of key forms: by name, by index, by unique ID



Phase 1:

Scripting Features

Object Model Support

- Promote consistency and intuitiveness

Get the color of the first picture

not

GetColor

- Lots of key forms: by name, by index, by unique ID
- **whose** clause counts



Phase 1:

Scripting Features

Object Model Support (Cont.)

- Custom verbs operate on referenceable objects (just like standard verbs)
- Most of work can be done with standard make, set, get verbs
- Include a rich set of properties
- Classes reflect conceptual user-oriented objects, not programmatic objects



Phase 1:

Scripting Features

Recordable

- Playing back should work, given same conditions as when recorded (be careful of omissions)
- Record commands that resemble what the user would write
- Don't sacrifice intuitive for recording
- Don't return references by unique ID when recording



Phase 1:

Scripting Features

Attachable

- Scripts menu
- Hierarchical menus
- Live updating of script list



Phase 1:

Scripting Features

Embeddable—embedding of scripts in the User Interface generally

- Associate scripts with interface elements
- User interface method of invoking is up to you



Phase 2:

Quality of Scripting Implementation

- How well does the app form a consistent and useful user scripting interface?
- 5 categories—12 points each (6—7 is average)



Phase 2:

Quality of Scripting Implementation (60 points)

- Design
- Completeness
- Transparency
- Stability
- Documentation



Phase 2:

Quality of Implementation

Design—fidelity to the application's structure and function

- Doesn't just mimic the graphical interface
- Faithful implementation of standard verbs and nouns
- Few custom verbs (events), many nouns (classes)
- Extends language consistently with standard terminology style



Phase 2:

Quality of Implementation

Completeness—depth, breadth,
and thoroughness

- Exposes *full* functionality
- Rich and extensive containment hierarchy—hook up everything
- Thorough, detailed, and complete



Phase 2:

Quality of Implementation

Transparency—resemblance to plain English words and syntax

- Users can write English sentences
- Common words are used for terms
- Produces natural, intuitive syntax
 - Non-programmers may get it right the first time
 - Experienced users can remember
 - If a programmer thinks a command is just a concept, it's good



Phase 2:

Quality of Implementation

Stability—absence of bugs

- Bug free
- Works as advertised
(according to dictionary)



Phase 2:

Quality of Implementation

Documentation—docs and examples provided

- Printed or electronic docs
- Example scripts
- Help facility
- Make the comments in your ‘aete’ explain sufficiently





Demo

Example Scorecard Review
Style—Marco Piovanelli

Example Review

Style 1.6 marco

Scriptability Scorecard Review Form

Product Name: Style
Version: 1.6
URL: <http://www.merzware.com/style/>
Category: text editor

Developer: Marco Piovanelli
Developer Email: support@merzware.com

Review Date: 4/29/99
Reviewer Name: Bill Cheeseman
Reviewer Email: cheeseb@mediaone.net

Phase 1-Scripting Features Supported

Object Model Support (0 to 25): 24/28
--extensive and natural set of objects, with sensible inheritance and properties, covering all aspects of the application; remarkably full support for whose clauses with complex capabilities and very fast execution; implements a 'properties' property for one-event formatting of text objects; character object class isn't double-byte aware

Recordable (0 to 5): 4/3
--very good, but doesn't record typing, making a text selection or drag-and-drop editing

Attachable (0 to 5): 5/5
--scripts menu supports hierarchical submenus and adds items on the fly; uses a script icon instead of "Script" (I chose not to deduct 2 points for this)

Embeddable (0 to 5): 5/3
--a script can be attached to the application or to a document using AppleScript or by placing a "Startup" script in the application folder; handlers can be attached to any menu command and to any scriptable user action (such as "make new document"); it is even possible to attach handlers that create new events which can be triggered from outside Style, so that Style's scripting interface is itself extensible

Phase 2-Quality of Scripting Implementation

Design (0 to 12): 12/12
--this class shared with Tex-Edit Plus is inventive and useful for scripting
--"one event" feature allows use of a single event



Phase 1:

Example Review

| | | |
|----------------------|-----------|----|
| Object model support | 20 | 25 |
| Recordable | 3 | 5 |
| Attachable | 5 | 5 |
| Embeddable | 2 | 5 |
| Total | 30 | 40 |



Phase 2:

Example Review

| | | |
|---------------|----|----|
| Design | 12 | 12 |
| Completeness | 9 | 12 |
| Transparency | 10 | 12 |
| Stability | 11 | 12 |
| Documentation | 8 | 12 |
| Total | 50 | 60 |



Summary:

Example Review

| | |
|--------------------|----|
| Scripting Features | 30 |
| Quality | 50 |

Score

80/4.0



AppleScript Scorecard URL:

www.mainevent.com/scorecard/



Terminology Tips



Terminology Tips

- Proxies for explicit default containment

Current record



Terminology Tips

- Proxies for explicit default containment

Current record

- Object accessors for these can be tricky:

Current record of current database

<property> of <property> of <object>



Terminology Tips

- Proxies for explicit default containment

Current record

- Object accessors for these can be tricky:

Current record of current database

<property> of <property> of <object>

- Document property inheritance:

<Inheritance>



Converting to Better Implementation

- Scrap non-object model implementation
- Update to better terms (facelift)



Steps to Scriptability

- Write down sentences (users also!)
- Try talking to your computer
- Extract the verbs and nouns
- Build a rough object model and 'aete'
- Write sentences that use the model
 - Have scripters try it too
- Iterate
- Write Code



Designing the Object Model Verbs vs. Nouns

- Users think in actions, but remember objects
- Balance the object-oriented user interface and the action-oriented scripting interface
- Implementing **set** and **get** for properties and objects is the most important part of your scriptability



Your Dictionary Is Your Specification

- Don't write code first and then decide what to call things—do the vocabulary first
- Don't focus on Apple Event communication—do focus on user scriptability
- Don't make scriptability a subset—design it all!



Issues of Style

- Use lowercase letters

choose printer, not **Choose Printer**



Issues of Style

- Use lowercase letters
choose printer, not **Choose Printer**
- Separate your words
Ask before overwriting, not
pAskBeforeOverwriting



Issues of Style

- Use lowercase letters
choose printer, not **Choose Printer**
- Separate your words
Ask before overwriting, not
pAskBeforeOverwriting
- Understand and use enumerations
Round up / down / to zero / towards nearest



Issues of Style

- Use lowercase letters
choose printer, not **Choose Printer**
- Separate your words
Ask before overwriting, not
pAskBeforeOverwriting
- Understand and use enumerations
Round up / down / to zero / towards nearest
- Declare record definitions



Terminology Tips

- Use terms consistent with other apps

Current document, not **active document**



Terminology Tips

- Use terms consistent with other apps

Current document, not **active document**

- Don't start property names with verbs

headers included okay

includes headers better

including headers best

not **include headers**



Declaring Record Definitions

- Useful for several things
 - Collections of information
 - Preference settings



Declaring Record Definitions

- Useful for several things
 - Collections of information
 - Preference settings
- Define a “fake” class in Type Definitions (suite ‘tpdf’)



Declaring Record Definitions

- Useful for several things
 - Collections of information
 - Preference settings
- Define a “fake” class in Type Definitions (suite ‘tpdf’)
- Properties of this class represent the labels of the record elements

Get the person info of employee 75

→ {name:"Cal", phone:"555-1212"}



AppleScript and Java

Special considerations

- Reorganize your suites
- Remove terminology events from 'aete'
- Remove some UI stuff generated by AWT
- Remove extraneous methods —
e.g., scripters can use properties
set position of <reference>
- Create real comments! (BeanInfo)



What You Can Do

- Read develop #24 and #21
- Do the object model!
- Consult an AppleScript expert
- Look at other applications for guidance
- Help get other developers to do things right



AppleScript Scorecard URL:

www.mainevent.com/scorecard/





99 | Worldwide
Developers
Conference

Discussion



Think different.TM



Welcome

To Advance through Presentation
Use Page Up and Page Down Keys



99 | Worldwide
Developers
Conference