

В А Ш П О М О Щ Н И К
AppleScript

Версия 2, 2002 г.

Содержание

1. Задачи, решаемые с помощью AppleScript	3
2. Краткое описание AppleScript?	4
Переменные и константы	5
Преобразование классов	6
Выражения	7
Сохранение	8
Русский язык	8
Словари	9
Дополнения (Scripting Additions)	11
3. Инструменты	12
Для написания, редактирования и отладки	12
Script Editor	
Script Debugger	
Scripter	
FaceSpan	
PreFab Player	
AppleScript Studio	
Для запуска сценариев	14
OSA Menu	
iDo Script Scheduler	
Script Menu	
4. Программы поддерживающие AppleScript	15

1. Задачи, решаемые с помощью AppleScript

Сколько раз вам приходится повторять одни и те же действия? При сортировке файлов, почты, форматировании текста, подготовке отчетов? Можете ли вы представить, например сколько надо совершить последовательно-повторяющихся действий наборщику или верстальщику, которому необходимо в программе телевидения выполнить выделения времени каждой передачи? Попробуем подсчитать: допустим разворот вмещает семь основных каналов ТВ; время вещания с 7⁰⁰ до 2⁰⁰ – 19 часов; в среднем одна программа длится примерно 30 минут; 7 дней недели. Итого 1862 раза необходимо подвести курсор к началу строки, выделить число, выполнить выделение, плюс к этому отформатировать названия каналов. А сколько понадобится времени верстальщику после сдачи номера на сортировку и архивирование файлов с готовой версткой, текстовыми и иллюстрационными оригиналами? По исследованию проведенным независимой исследовательской компанией Gistics в 1998 году, поиск каких-либо файлов в архивах, не организованных должным образом, заканчиваются неудачей в 7 случаях из десяти. И такие проблемы присущи многим другим видам деятельности, а не только редакционно-издательским процессам при обработке иллюстративных и текстовых массивов данных. А объем информации, который приходится обрабатывать, растет и уменьшаться, скорее всего, не собирается. Не проще ли доверить рутинную работу AppleScript, а тем временем насладиться чашечкой ароматного кофе (если рядом нет еще одного компьютера)?

Но как и любой другой язык, AppleScript перед использованием придется сначала изучить. Однако этого не стоит пугаться, на данный момент AppleScript самый простой и удобный язык. Его конструкции максимально приближены к конструкциям английского языка. Слова имеют вид существительных, глаголов, прилагательных, предлогов и т.д. Изучив и получив первые результаты его использования, вам станет очевидно, что вы не потратили время зря. Развивая свои знания в области AppleScript, вы со временем сможете комбинировать, собирать, как из кубиков, подпрограммы, целые сценарии или их части, для составления более сложных программ, направленных на решение ваших задач.

Использование AppleScript для построения системы управления цифровыми данными может сэкономить много ваших средств. Важно то, что AppleScript может работать со многими программами, которые у вас уже есть, существенно расширяя их возможности. AppleScript в силах превратить ваши разобщенные программы в единый мощный комплекс управления цифровыми данными. А что же готовые комплексы управления цифровыми данными, архивами? Продукты низкой стоимости, такого класса как каталогизаторы, не многих смогут удовлетворить своими ограниченными возможностями. Более дорогие продукты – стоимостью уже порядка десятков тысяч долларов – с широкими областями применения, множеством параметров настройки многим могут быть не по карману, к тому же в их стоимость следует включить сервисное обслуживание этих систем и обучение персонала. Причем затраты будут возрастать при увеличении численности пользователей работающих с этой системой.

Помимо управления данными, сценарии можно использовать в чисто утилитарных целях. (Хотя, во многом, такие сценарии остаются инструментом для управления данными.) Помогут работе с компьютером всевозможные сценарии, комментирующие или изменяющие тип (и/или тип создателя) большой группы файлов, управляющие размещением файлов на диске, управляющие контрольными панелями, принтерами, монтирующие сетевые диски, создающие папки по текущей дате... Таких примеров можно приводить много. Создавайте! Все зависит только от вашей фантазии и программ, которые поддерживают AppleScript.

Сценарии можно использовать также и для развлечения. Представьте себе сценарий, раз в день, при включении машины, меняющий фоновый рисунок на вашем экране, выбирая изображения из заданной папки.

AppleScript позволяет использовать уже привычные пользователю программы для выполнения новых, более комплексных задач. Язык AppleScript достаточно прост и человеку, знакомому с английским языком не составит большого труда изучить его.

AppleScript является бесплатным и поставляется с MacOS. Так же с MacOS поставляется и редактор сценариев Script Editor, который позволяет проверять, компилировать и даже записывать сценарии из некоторых программ (записываемые программы). О других инструментах для работы со сценариями будет рассказано ниже.

2. Краткое описание AppleScript?

AppleScript похож на многие другие языки сценариев, языки программирования, макро-языки. Вы найдете синтаксис AppleScript очень похожим на обычный английский язык, это делает его легким в изучении и использовании.

AppleScript динамический, объектно-ориентированный язык написания сценариев. Ключевая особенность – способность посылать команды объектам во многих различных приложениях, включая непосредственно Mac OS. Объект – это элемент, такой как файл или папка в окне Finder, слово или параграф в текстовом редакторе, который может ответить на команды посылаемые сценарием. AppleScript динамически определяет – всегда, когда это необходимо – какие объекты и команды программа распознает, основываясь на информации хранимой в каждом скриптуемом (scriptable) приложении. Команда – это слово или серия слов, используемых в выражениях AppleScript для выполнения какого-либо действия. Каждая команда направлена на цель, объект, отвечающий на команду. Целью обычно является объект приложения.

Apple Script не воспроизводит записанную последовательность действий клавиатуры и мыши с AppleScript вы обращаетесь прямо к объектам программы. Объектами могут быть: слово в текстовом документе, бокс в QuarkXPress, запись в базе данных File Maker, файл, папка и многие другие элементы операционной системы и приложений. Однако некоторые программы позволяют записывать действия, производимые над объектами. Одной из таких программ является Finder. Попробуйте, нажмите кнопку Record в Script Editor и, перейдя в

Finder, перемещайте файлы и папки, открывайте папки и перемещайте их окна по экрану. Ваши действия будут записываться в виде команд Apple Script в документе программы Scripteditor. Действуя подобным образом, написать нормальный сценарий не удастся, зато можно получить представление о взаимосвязи команд и объектов Apple Script с элементами, которые предоставляют программы.

Переменные и константы

Переменные это данные, которые могут быть представлены, присвоены, и которыми можно управлять в скриптах. Каждая переменная соответствует классу переменных (**value class**), которые являются категорией, описывающей какой тип данных присваивается переменной и какими операторами она может быть использована. Классы во многом схожи с типами данных переменных в других языках. Классы могут быть двух категорий: простые переменные и структурированные переменные.

Наиболее часто используемые простые классы: **string**, **integer**, **real**, **boolean** и **class**. Иногда значения, возвращаемые приложением (в ответ на команду), присваивается переменной класса **data**. Этот класс является сырыми данными и его можно только присвоить переменной, но нельзя представить в виде строки или числа.

Список – значения класса **list** – коллекция значений, расположенных в определенном порядке. Список очень похож на тип данных, называемый массивом, других языков программирования. Значения, содержащиеся в списке, называется элемент (**item**). Каждый элемент списка может быть любого класса. К элементам можно получить доступ по так называемым формам обращения (**Reference Forms**), например форма обращения по индексу –

```
item 3 of {"this", "is", "a", "list"}
```

определяет "a". Пустой список – список не содержащий ни одного элемента – представляется парой фигурных скобок {}.

Класс **record** (запись) представляет из себя не имеющую определенного порядка коллекцию свойств (**properties**). Каждое свойство имеет имя (**label**), и свойства различаются друг от друга именем. Обращаясь к имени свойства вы получаете доступ к его значению.

```
set MacUser to {name: "Vasia", age: 16, sex: "male"}
return name of MacUser
-- результат: "Vasia"
```

В записи не может быть свойств с одинаковыми именами. Вы не можете обращаться к свойствам по индексу как к элементам списка. Аналогично, объекты имеют коллекцию свойств, будь то объект AppleScript или объект из словаря приложения.

Очень удобен при работе с временем и датой класс **date**. Полностью значение **date** описывает день недели, дату и время. Если вы определите только часть этой информации, то AppleScript заполнит недостающие части значениями по умолчанию. Вы можете определять и получать свойства переменной **date**, которые

соответствуют различным частям информации о дате и времени. При использовании таких свойств этого класса, как месяц и день недели необходимо выполнить дополнительные преобразования для представления их в числовом виде или в виде строк с русскими названиями дней недели и месяца.

Константы являются классом `constant`. Значения класса констант – зарезервированное слово определенное в AppleScript или в словаре приложения. Вы не можете определять собственные константы.

Преобразование классов

AppleScript преобразует значения в одном из двух случаев:

- а) при использовании оператора `As`;
- б) автоматически, когда значение имеет отличный класс от того, который требует команда или операция.

Возможность изменять значение из одного класса в другой является встроенной функцией AppleScript или предоставляется командами дополнений (scripting addition).

Когда преобразуются значения `string` в `integer`, `real` или наоборот, AppleScript использует текущие установки в контрольной панели «Numbers» для десятичных и шестизначных чисел, чтобы определить какие разделители использовать в `string`.

Когда `string` преобразуется в `date` или наоборот, AppleScript использует текущие установки в контрольной панели «Date & Time».

Преобразование из	Преобразование в														
	Boolean	Class	Constant	Data	Date	Integer	International Text	List (один элемент)	List	Number	Real	Record	String или Text	Styled Text	Unicode Text
Boolean	■														
Class		■													
Constant			■												
Data				■											
Date					■										
Integer						■									
International Text							■								
List (один элемент)	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
List								■							
Real									■	■					
Record											■				
String													■	■	■
Unicode Text													■	■	■

Выражения

Для создания переменной ей присваивается значение. Для этого используются две команды: **set** и **copy**.

```
set MyName to "Vasia"
```

Вместо значения переменной можно присвоить ссылку на это значение командой **a reference to**.

```
set MyText to (get a reference to story 1 of text box 1)
```

В AppleScript доступны многие арифметические операции, логические операции **and**, **or**, **not**, операторы сравнения. Для соединения строк, списков и свойств используется оператор конкатенации **&**. Многие операторы имеют синонимы, что делает язык более удобным и интуитивно понятным.

Для контроля за ходом сценария используются управляющие операторы:

«Операторные скобки **Tell... End Tell**» определяют объект, которому будут отсылааться команды.

```
tell front window of application "AppleWorks"  
  close  
end tell
```

«Операторные скобки **If... Else... End If**» позволяют вам выполнять различные участки кода, в зависимости от одного или нескольких условий.

```
if button returned of result = "Yes" then  
  set audit to true  
else  
  set audit to false  
end if
```

«Операторные скобки **Repeat... End Repeat**» позволят вам повторять выполнение серии операторов.

```
repeat with n from 1 to numParagraphs  
  set paragraph n to (n as string) & " " & paragraph n  
end repeat
```

«Операторные скобки **Try... On Error... End Try**» позволяют обрабатывать ошибки.

«Операторные скобки **Considering... End Considering** и **Ignoring... End Ignoring**» позволят вам соглашаться с такими признаками, как регистр, пунктуация и ряд пробелов в сравнениях строк или игнорировать их.

Вы также сможете определить подпрограммы и функции для удобства работы с кодом сценария; определить свои собственные объекты сценария, которые имеют свойства и отвечают на команды.

Фильтры позволяют выбрать из группы объектов, объекты с каким-нибудь одинаковым свойством, например команда для **QuarkXPress**

```
get generic boxes whose color is "Black"
```

вернет список из ссылок на все боксы, чей цвет заливки черный.

Сохранение

AppleScript имеет несколько разных форматов для сохранения сценария. Для запуска сценариев из Finder, как обычное приложение, вам необходимо сохранить сценарий в формат Application. Для запуска сценариев с помощью таких инструментов, как OSA Menu Lite, iDo Script Scheduler вам может потребоваться сохранить их в формат Compiled Script (компилированный сценарий). В этом случае, при двойном щелчке по такому сценарию в Finder, он откроется в редакторе сценариев. Если по какой-либо причине вы не смогли откомпилировать сценарий, но вам нужно его сохранить, то любой редактор сценариев даст вам возможность сохранить сценарий в текстовом виде. Для защиты сценария от редактирования, сохраняйте его как Read-Only Compiled Script или Read-Only Application, после этой операции сценарий не может быть открыт для редактирования и его содержимое нельзя будет просмотреть и изменить.

Русский язык

Обычно идентификаторы в AppleScript должны начинаться с символа от «a» до «z», могут содержать как строчные, так и прописные символы латинского алфавита, числа (0–9), и символ подчеркивания (). Но, идентификатор заключенный в скобки из символа вертикальной линии (|) может содержать любой набор символов, кроме конечно символа «|». Это позволяет использовать идентификаторы на русском языке! Возможно AppleScript единственный из современных языков, который позволяет использовать русские идентификаторы! Однако, это обстоятельство, не мешает вам запутаться в собственном сценарии. Старайтесь давать такие имена переменным, чтобы при минимальном количестве символов, их назначение было понятно даже постороннему человеку.

При работе со строками русского языка могут возникнуть некоторые проблемы. При сравнении строк AppleScript считает символы «Ж» и «Я» одинаковыми. Использование соглашений diacriticals или expansion не дает ни каких изменений. Выход из этой ситуации можно найти в использовании команды ASCII number из стандартного дополнения, но неправильное использование этой команды сильно снизит скорость выполнения сценария.

Начиная с обновления MacOS 9.2.2 появилась еще одна проблема, которая не исправлена по сей день. Если в пути к файлу есть символы кириллицы. То использование класса alias становится невозможным.

```
set MyFile to alias "Macintosh HD:Отчеты:Продажи.doc"
```

```
-- ошибка компиляции: файл "Macintosh HD:Отчеты:Продажи.doc" не найден
```



Если какая-либо команда приложения или дополнения возвращает значения класса alias, постарайтесь преобразовать их в string или ссылку. Например обработчик события open передает в качестве параметра список значений alias. Обработка события open используется для создания дроплета (droplet), который запускается натягиванием файлов или папок на иконку сценария.

```
on open (theList)
```

```
set NewList to {} -- создаем пустой список
```



```

tell application "Finder" -- команды будет принимать Finder
    repeat with OneItem in theList -- запускаем цикл,
        -- с каждым повтором цикла переменной OneItem присваивается
        -- значение очередного элемента списка
            set end of NewList to item OneItem -- ставим в конец
            -- нового списка значение OneItem но уже в виде ссылки
            -- на файл или папку
        end repeat
    end tell
end open

```

Словари

Во многом существование AppleScript определяется поддержкой его большим количеством различных программ и дополнений. Среди этих программ, учитывая склонность Мака к настольным издательским системам: QuarkXPress, Adobe InDesign, Adobe Photoshop, Adobe Illustrator. Также AppleScript поддерживают большинство популярных приложений, в число которых входят FileMaker, BBEdit, Textedit, Apple Works, iViewMultimedia, все программы пакета MSOffice, и множество других, используемых вами в повседневной работе. Программы, которые могут отвечать на команды AppleScript, называются скриптуемыми (scriptable). Программы для Макинтош очень разнообразны, от текстовых редакторов до редакторов звука и видео. Конечно, все программы имеют различные интерфейсы, другими словами каждая программа говорит с пользователем на своем диалекте. Для того чтобы узнать, что понятно программе, не придется долго искать ключевые слова по файлам справки – вся информация находится в скриптуемой программе. Список ключевых слов, известных программе, так и называется – словарь (dictionary). Словарь содержит имена команд, объектов и их свойств. Открывая в редакторе сценариев через команду меню «Open Dictionary» файл какой-либо программы, вы получите доступ к словарю.

В редакторе сценариев словарь отображается в виде двух панелей. Слева дается список ключевых слов из команд, объектов и классов. Ключевые слова разделены на группы (suits), названия которых выделяются жирным. В самом начале в большинстве случаев располагается Required Suite и Standart Suite. Required Suite часто указывают и оставляют пустым. **Все приложения без исключения поддерживают команды Run, Open, Quit, Print**, поэтому их описание не приводится в словарях. Например, команду Quit удобно использовать для завершения даже тех приложений, которые не поддерживают AppleScript. Если эти команды переопределены, то они сгруппированы в Standart Suite. В зависимости от программы и структуры словаря можно встретить группы: Text Suite, Internet Suite и другие.

Среди команд объекты и классы выделяются курсивом. При выборе ключевого слова в левой панели справа отображается его описание. Для команды описываются параметры, которые с ней передаются и возвращаемый результат. Для того, чтобы команды в словарях согласовывались с концепцией языка – быть похожим на человеческий язык – перед параметрами ставятся самые разнообразные ключевые слова.

`display dialog "текст диалога" with icon stop`
(with icon – ключевое слово, предшествующее параметру stop).
Параметры, замкнутые в квадратные скобки, являются необязательными.

Для объектов в описании приводятся: множественное число, список элементов и, наконец, свойства объекта. Множественное число используется для обращения ко всем доступным объектам данного типа и изменения каких-либо их свойств сразу у всей группы. В английском диалекте AppleScript "множественное число" образуется по правилам английского языка.

```
delete generic boxes of page 1 of document 1 of application "QuarkXPress"
```

Синонимом для множественного числа является ключевое слово every. В указанном примере generic boxes можно заменить на every generic box.

Каждый объект может содержать в себе другие объекты, например: слово содержит символы; программа содержит окна и документы. Такие вложенные объекты называются элементами (elements) для содержащего их объекта. В комментарии к элементам указано как можно выделить в объекте один элемент из множества других: по имени, по относительному положению, по идентификатору.

Важная часть объекта – его свойства (properties). Именно через свойства получают сведения о состоянии объектов. Изменяя или устанавливая какие-либо значения свойствам воздействуют на объект, придавая ему те или иные характеристики, изменяя его поведение в программе. Свойство объекта character, называемое style, может быть установлено на plain, bold, outline или другое (применимо для многих приложений, в которых предусмотрена обработка текста). Свойство bounds объекта window (Finder и другие приложения) описывает положение этого окна в виде списка с целыми числами – координатами – {40, 60, 140, 160}. Также свойства важны для идентификации объектов. Для того чтобы выделить из множества однотипных объектов один, в сценариях определяют значение свойств name, index, ID. В различных программах имена этих свойств могут отличаться. Name определяет заранее заданное имя объекта. Например в QuarkXPress многим типам объектов можно задать имя через свойство name. Пользователю имя данное объекту не видно и не доступно для изменения, оно задается только с помощью в AppleScript. Но однажды задав имя, можно будет легко найти этот объект из всего множества линий, боксов, текстовых боксов, страниц, документов QuarkXPress. Другое дело Finder – в нем имена объектов: Item, File, Folder, Application одинаковы и на экране, и в AppleScript.

Index определяет порядковый номер объекта и может изменяться при создании или удалении объекта. Некоторые приложения присваивают объектам уникальные идентификаторы, они зачастую хранятся в свойстве с именем ID класса number, и доступны только для чтения. Доступным только для чтения свойствам нельзя устанавливать какие-либо значения, но их значения можно присваивать переменным или использовать в качестве операнда. В описании рядом с такими свойствами стоит отметка [r/o].

Дополнения (Scripting Additions)

Дополнения дают возможность использовать дополнительные команды или преобразование классов в сценариях. Файлы дополнений должны быть расположены в папке Scripting Additions (в System Folder), чтобы AppleScript получил доступ к дополнительным командам. Один файл дополнения может содержать несколько команд. Например, Standard Additions распространяемый с AppleScript, включает команды для работы с клипбордом, использования речевых функций, отображения стандартных диалогов сообщений, выбора файлов и папок, и многое-многое другое. Используя команды дополнений нет необходимости указывать объект-цель, если сценарий управляет компьютером на котором он запускается. Когда сценарий управляет другой машиной включенной в сеть, то естественно указывается объект-цель.

```
tell application "Finder" of machine "Vasia"
    say "Privet!"
end tell
```

Стандартные дополнения распространяемые с AppleScript дают возможность управлять многими компонентами системы: для управления принтерами используется Desktop Printer Manager; для управления Open Transport и Remote Access используется Network Setup Scripting; FileSharing Commands позволит управлять соответствующими функциями без открытия контрольной панели; Keychain Scripting и т. д. Для информации о стандартных дополнениях распространяемых с AppleScript смотрите раздел AppleScript в Mac OS Help Center, или веб-сайт AppleScript www.apple.com/applescript/.

AppleScript имеет ограниченные возможности по взаимодействию с пользователем посредством диалогов. Преодолеть эти ограничения призваны дополнения Dialog Director и Appearance OSAX. Кроме того, эти дополнения позволяют создавать окна различных типов, окна с линейкой хода процесса (Progress Bar).

Dialog Director бесплатное дополнение. Его команды можно разделить на два типа: первые, позволяют сократить время на создание и расположение на экране диалогов, вторые дают полный контроль над созданием диалогов, и потому имеют более сложную структуру. При использовании команды dd auto dialog наблюдались такие нарушения в работе системы, как прекращение обновления экрана.

Appearance OSAX распространяется в виде ограниченного-по-времени дополнения. Позволяет работать с ним 12 дней без регистрации. Имеет удобные команды для создания окон и диалогов. Проблем замечено не было.

За время существования AppleScript различными разработчиками выпущено очень много дополнений, выполняющих разнообразные функции. Многие из них можно найти на сайте www.osaxen.com/index.php3

3. Инструменты

Для написания, редактирования и отладки

Редактор сценариев Script Editor (Apple Computer, Inc.) поставляется с MacOS, который позволяет проверять, компилировать и записывать сценарии из некоторых программ. Script Editor позволяет просмотреть ход программы и результаты действия команд. Можно просмотреть объекты и команды предоставляемые приложениями и дополнениями. Поддерживает сценарии длиной до 32 Кбайт. В Script Editor отсутствуют инструменты поиска и замены. Размещение – Ваш загрузочный диск:Applications (Mac OS 9):Apple Extras:AppleScript:Script Editor"

Начиная работать с AppleScript вам необходимо ознакомиться со Script Editor. В отличие от просто текстового редактора Script Editor имеет некоторые особенности, отвечающие его назначению. В основу прочих редакторов для AppleScript положены те же принципы что и в Script Editor.

Каждое окно с текстом сценария оснащено управляющими кнопками и областью для комментария к сценарию или описания его работы. Три кнопки слева: запись команд (эффективно только с приложениями, которые поддерживают запись), остановка выполнения и запуск. Справа размещена кнопка проверки синтаксиса. После ввода нового текста кнопка становится доступной и при нажатии компилирует сценарий, заодно производя форматирование. После удачной компиляции каждая строка отбивается табуляцией в соответствии с вложенностью операторов. Каждое слово приобретает стилевую разметку, которая устанавливается в диалоге «AppleScript Formatting». Можно задавать различные шрифты, их начертания, размер и цвет для команд языка и программ, переменных, комментариев, значений и ссылок.

Если вы набираете русский текст сценария в Script Editor и он отображается некорректно - измените шрифт для некомпилрованного текста на «Geneva CY» или «PriamojProp». Используйте различные цвета для выделения различных выражений - текст сценария будет легче читаться.

Для отладки сценариев в Script Editor применяются окна «Event Log» и «Result», которые вызываются из меню или клавишами `cmd+L` и `cmd+E` соответственно. В окне «Event Log» отображаются команды языка или программ, вызываемые сценарием во время выполнения, и результат их действия. Для отображения значения переменных в ходе выполнения сценария используется оператор `log`. В окне «Result» отображается результат последней команды сценария или значение возвращаемое оператором `return`, который прекращает работу сценария.

Пример:

```
set myString to "Macintosh"  
log myString -- в окне "Event Log" ("Macintosh")  
return myString -- результатом "Macintosh" в окне «Result
```

Script Debugger (LateNightSoftware)

Когда скромных возможностей Script Editor станет недостаточно, стоит ознакомиться со Script Debugger производства LateNightSoftware. Script Debugger не бесплатная, но исключительно мощная среда для разработки, редактирования и отладки сценариев. Редактор полностью поддерживает технологию «перетаскивания». Основная особенность Script Debugger, оправдывающая его название – возможность пошаговой отладки сценариев. Отладка дроплетов «превращается в удовольствие» – программа может вызывать события open, а также idle и quit для сценария находящегося в окне редактирования. Многие вспомогательные функции Script Debugger'a построены в виде плавающих палитр. Так функции запуска сценариев с передачей команд open, idle и quit являются частью палитры «Scripts». В этой палитре можно найти еще много полезных инструментов для управления окнами, упрощения редактирования, все они построены в виде сценариев Script Debugger'a и при желании, к уже существующим, можно добавлять и свои вспомогательные сценарии. Фрагменты кода можно хранить в палитре «Clippings». Открывая словари приложений, путь к ним сохраняется в палитре «Applications». Словари Script Debugger способен отображать в трех режимах: в обычном; как объектную иерархическую модель; в режиме Explorer, где отображаются все текущие объекты программы, их элементы и свойства с текущими значениями, которые здесь же можно и изменить. Explorer крайне удобен для изучения объектной модели приложений. Помимо перечисленного в Script Debugger имеется еще много интересного и полезного.

www.latenightsw.com

Scripter (Main Event)

Одна новая особенность – обещают создатели Scripter 2.5 – окно Object Prober, отображающий иерархическую структуру объектов, и позволяющее видеть все, присутствующие в запущенной скриптуемой программе объекты, и видеть свойства текущего объекта. Другая новинка – Search Terminologies, которое позволяет искать имена команд, свойств, констант в словаре дополнения или какого-либо скриптуемого приложения. В остальном эта среда имеет хорошие возможности для отладки сценариев, для редактирования и хранения фрагментов кода, однако отстает от Script Debugger в удобстве и интуитивности интерфейса.

www.mainevent.com

FaceSpan (Digital Technology International)

Прикладной пакет FaceSpan, позволит вам создавать сложные графические пользовательские интерфейсы для сценариев AppleScript. Он включает в себя редакторы для написания сценариев, конструирования окон и меню, также как и среду управления, посредством которой вы сможете проверять, редактировать, дублировать, удалять каждый компонент интерфейса или ресурс вашей будущей программы.

www.facespan.com

PreFab Player (PreFab Software)

При помощи PreFab Player – расширения для работы со сценариями, которое средствами AppleScript обеспечивает доступ к пользовательскому интерфейсу практически любой программы – ко многим приложениям можно добавить функции AppleScript.

www.prefab.com

AppleScript Studio для MacOS X – новый исключительно мощный инструмент для создания полноценных Cocoa-приложений с интерфейсом Aqua.

Для запуска сценариев

В комплекте с системой MacOS 9 идут две утилиты. Они называются OSA Menu Lite и IDo Script Sheduler. Если вы намерены пользоваться AppleScript повседневно, то немедленно, после прочтения этой статьи, устанавливайте их. Первая – позволяет запускать сценарии из меню или нажатием комбинации клавиш. Вторая – как следует из названия, запускает сценарии по созданному вами расписанию. Обе программы в тех версиях, что идут с MacOS, запускают только сценарии сохраненные как компилированный сценарий (Compiled Script). Однако, полная версия OSA Menu бесплатна, также как и OSA Menu Lite (*www.lazerware.com*). При этом в OSA Menu можно установить режим, при котором сценарий выполняется как одна команда – выполнение в несколько раз быстрее, чем обычно, не некоторые сценарии не будут работать в этом режиме.

OSA Menu (Lazerware Software)

Это иконка-меню (подобная меню Application), которая появляется с правой стороны полосы меню. Элементами меню являются сценарии: выберите элемент меню и сценарий запустится. Некоторые команды доступны всегда, а некоторые становятся доступными только в пределах определенных программ. Основное достоинство размещения сценариев в меню то, что вы можете их вызывать одним щелчком мыши, не покидая текущего приложения. Есть возможность назначать сценариям комбинации клавиш. Для этого в конце имени ставится символ \, далее следует символ клавиши, после знака плюс можно установить клавиши модификаторы O – option, S – shift, C – control. Можно назначить даже F-клавиши или другие несимвольные клавиши, но через некоторое время могут последовать сбои.

Сценарии размещаются в папке «Scripts» в вашем System folder, и каждое приложение получает свою собственную папку «Scripts» со сценариями предназначенными для этого приложения, точно также как и размещение сценариев в папке «Universal Scripts» делает их доступными во всех приложениях.

www.lazerware.com

iDo Script Scheduler (Sophisticated Ciurcuits, Inc)

Когда вы не за компьютером, iDo Script Scheduler будет запускать сценарии, в любое заданное вами время. Используйте iDo Script Scheduler чтобы создать многократно повторяющиеся «события». Каждое событие может запустить сценарий AppleScript, и передать ему строку параметров. iDo Script Scheduler поддерживает многопользовательскую архитектуру Mac OS 9. Каждый пользователь может создавать различные расписания, которые будут запускаться только тогда, когда, этот пользователь зашел в систему.

Когда вы в первый раз установите iDo Script Scheduler, он запускается в режиме «Lite». Вы можете создать не более трех событий используя любые временные триггеры. Чтобы получить большего от iDo Script Scheduler, или использовать его в коммерческих целях, вы можете за дополнительную плату проапгрейдить его до режима «Enhanced». Режим Enhanced позволит вам создавать неограниченное количество событий и добавит два мощных триггера: горячая клавиша и простой системы.

www.sophisticated.com

Script Menu для MacOS X - ищите по адресу: *www.apple.com/apple-script/macosx/script-menu/*.

4. Программы поддерживающие AppleScript

Скриптуемые программы можно разделить на записываемые и незаписываемые. Записываемые программы – такие как упоминавшееся выше Script Debugger или тот же Finder – передают для Script Editor информацию о действиях пользователя. Script Editor, после нажатия кнопки запись, записывает их в виде последовательности AppleScript команд. Записываемых программ не много, всего около двух десятков, и потребность в использовании записи возникает редко, но она может помочь начинающим освоить AppleScript.

Далее скриптуемые программы можно условно разделить на три группы по поддержке команд AppleScript:

- а) минимальная (поддерживают только команды Do Script или Required Suite);
- б) функциональная (поддерживают Core Suite для сохранения, печати, и т. д.);
- в) полная (поддержка полной объектной модели и особенностей программы).

К первой группе относятся некоторые продукты Adobe такие, как Adobe Photoshop до версии 6.0, Adobe Illustator 8. Данные программы позволяют запускать только последовательность действий. Возможности Photoshop (5.0 – 6.0) можно существенно расширить используя plug-in PhotoScripter (Main Event, *www.mainevent.com*), открывающий для доступа из сценария к обширной объектной модели растрового редактора. Теперь главные приложения фирмы Adobe поддерживают AppleScript в полной мере с помощью

бесплатных плагинов, которые можно скачать с их сайта www.adobe.com.

CorelDraw 8 позволяют автоматизировать многие действия программы. Кроме того, пользователь может записывать из CorelDraw 8. Есть одно ограничение: вместо модели объектов AppleScript здесь используются команды, которые имитируют собственный пользовательский интерфейс приложения.

В пакете Macromedia FreeHand версий 7 и 8 поддержка AppleScript ограничена в основном теми функциями программы, которые ориентированы на пакетную обработку, такими как экспорт файлов и операции поиска и замены.

ScriptMaster XT (Jintek, www.jintek.com/scriptmaster.html) дополнит и без того богатую управляемость программы верстки QuarkXPress плавающей палитрой с возможностью записывать и запускать сценарии не покидая основной программы, а также имеет набор готовых сценариев.

Для создания масштабных проектов, управляющих машинами и файлами рабочей группы или целого отдела с помощью AppleScript, будет незаменим FileMakerPro. При небольшой объектной модели он, все равно, позволит получить доступ к своим объектам. Кроме того, FileMakerPro может хранить, компилировать и запускать сценарии из полей базы данных, из шагов его собственных сценариев.

Список скриптуемых, записываемых, и присоединяющих программ:

www.apple.com/applescript/enabled.00.html

Еще один достаточно устаревший список скриптуемых программ:

www.tiac.net/prefab/scriptweb/scriptableapps.html