

В А Ш П О М О Щ Н И К

AppleScript

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 полоса 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 файлов с готовой версткой, текстовыми и иллюстрационными оригиналами? По исследованию проведенным в 1998 году, которое спонсировала компания Apple Computer оказалось, а проводила независимая исследовательская компания Gistics, 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 обработки 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, 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 возможностями настройки многим могут быть не по

карману, так как 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 привести много, все зависит только от вашей фантазий 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. 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. Если вам приходилось использовать какой-либо из этих языков, то вы найдете синтаксис AppleScript очень похожим на обычный английский язык, что 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.

Переменные и константы

Переменные это данные, которые могут быть представлены, присвоены, и которыми можно управлять в скриптах. Каждая переменная соответствует **классу переменных (value class)**, который является 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, но есть и много особенностей, присущих только AppleScript. 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.

Список — значения класса **list** — коллекция значений, расположенных в определенном порядке. Список очень похож на тип данных, называемый массив, 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, походит на тип данных "запись" других языков программирования. Представляет из себя не имеющую определенного порядка коллекцию свойств. Аналогично свойствам объектов приложений, каждое свойство имеет имя (**label**), и свойства различаются друг от друга именем. В записи не может быть свойств с одинаковыми именами. Вы не можете обращаться к свойствам по индексу как к элементам списка.

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) в ответ на использование 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	Одноэлементный List	Многоэлементный List	Number	Real	Record	String или Styled Text	Unicode Text
Boolean	■						■							
Class		■					■							
Constant			■				■					§		
Data				■			■							
Date					■		■					■		
Integer						■	■			■		■	■	
International Text							■					■*	■	■
Одноэлементный List	■	■	■	■	■	■	■	■	■	■		■	■	
Многоэлементный List								■				†		
Real						‡	■			■	■	■	■	
Record							■	■			■			
String						■	■	■		■		■	■	■
Unicode Text						■*						■*	■	■

* Некоторая информация может быть потеряна при выполнении данного преобразования.

† Только список, все элементы которого могут быть преобразованы в строку, может быть преобразован в строку.

‡ Только действительное число, у которого нет дробной части, может быть преобразовано в целое.

§ Появилось в AppleScript версии 1.3.7.

Выражения

Для создания переменной ей присваивается значение. Для этого используются две команды: set и сору. Вместо значения переменной можно присвоить ссылку на это значение командой a reference to.

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 или игнорировать такие признаки, как регистр, пунктуация и ряд пробелов в сравнениях строк.

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 боксов, чей цвет заливки черный.

Подробную информацию о операторах AppleScript вы найдете в «AppleScript Language Guide».

Сохранение

AppleScript имеет несколько разных форматов для сохранения сценария. Для запуска сценариев из Finder, как обычное приложение, вам необходимо сохранить сценарий в формат Application (в новом ScriptEditor — Classic Applet или Mac OS X Applet, в зависимости от

того где ваш сценарий будет работать). 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.

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 макро-языков, который позволяет использовать русские идентификаторы! Однако, это обстоятельство, не мешает вам запутаться в собственном сценарии.

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.

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 к дополнительным командам. Один файл дополнения может содержать несколько команд. Например, **Standard Additions** распространяемый с AppleScript, включает команды для работы с клипбордом, получения пути к файлу, использования речевых функций, отображения стандартных диалогов, диалогов выбора файлов и папок, и многое другое. Используя команды дополнений нет необходимости указывать объект-цель.

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 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 «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

Бесплатные: Script Editor, OSA Menu Lite

Редактор сценариев **Script Editor** (Apple Computer, Inc.) поставляется с MacOS, который позволяет проверять, компилировать и даже записывать сценарии из некоторых программ. Script Editor позволяет просмотреть ход программы и результаты действия команд. В нем можно задать различный стиль текста для команд, переменных, ссылок. Можно просмотреть объекты и команды предоставляемые приложениями и дополнениями. Поддерживает сценарии длиной до 32 килобайт. В Script Editor отсутствуют инструменты поиска и замены. Последние версии ведут себя странно, подтормаживая при копировании и вставке текста.

OSA Menu Lite (Leonard Rosenthol) поставляется с MacOS.

OSA Menu — 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.

В основном действия сценария направлены на активное приложение, и наиболее используемы сценарии работающие с объектами выделенными пользователем. Сценарии в OSA Menu выглядят подобно расширенному набору команд приложения. Основное достоинство размещения сценариев в меню то, что вы можете их вызывать одним щелчком мыши, не покидая текущего приложения.

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

Условно-бесплатные: iDo Script Scheduler

iDo Script Scheduler позволяет вам запускать сценарии в любое время, когда вы захотите. Вы можете использовать его, чтобы наделить ваш Макинтош такими задачами, как прием почты каждый вечер, или резервировать важные данные, когда вы не за компьютером.

Используйте iDo Script Scheduler чтобы создать многочисленно повторяющиеся «события». Каждое событие может запустить сценарий AppleScript, и передать ему строку параметров.

iDo Script Scheduler также поддерживает многопользовательскую архитектуру Mac OS 9. Каждый пользователь может создавать различные графики, 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 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

Коммерческие: **Scripter®**, **Script Debugger**, **FaceSpan**

Хорошей заменой для Script Editor будет , награжденный наградой Macworld Editor's Choice среда **Script Debugger** (Late Night Software). Он предоставляет дополнительные возможности, которые вы найдете неоценимыми при разработке сценариев. Script Debugger обеспечивает AppleScript полноценной средой разработки и отладки. Он имеет дополнительные возможности для редактирования текста, включая поддержку сценариев размером более 32 Кбайт, поиск с заменой, drag-and-drop. Script Debugger позволяет перемещаться по разделам сценариев за один шаг, проверять переменные вашего сценария во время выполнения. Script Debugger также предоставляет инструменты для просмотра команд и объектов дополнений и скриптуемых приложений, отображающие их в виде иерархической структуры или в виде списка с заполнением текущими значениями. Script Debugger скриптуем. Он допускает запись действий пользователя. Путем подключения сценариев в меню, хорошо настраивается под нужды и привычки пользователя. Использует систему «клиппирования» (от англ. — *clipping*) для создания библиотек из фрагментов кода.

www.latenightsw.com

Одна новая особенность — обещают создатели **Scripter 2.5** (Main Event) — 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 интуитивности интерфейса.

www.mainevent.com

Прикладной пакет **FaceSpan** (Digital Technology International), 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

При помощи **PreFab Player** (PreFab Software) — 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

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 расширить возможности новичков и помочь им 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, Adobe Illustrator 8. Данные программы позволяют запускать только последовательность действий. При выходе Adobe Illustrator 9 заявлялось, что он будет более широко поддерживать AppleScript...?... Возможности Photoshop можно существенно расширить используя plug-in PhotoScripter (Main Event, *www.mainevent.com*), открывающий для доступа из сценария к обширной объектной модели растрового редактора.

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