

Автоматизируем рабочий процесс: сценарии в Photoshop CS

Как вы, наверное, уже поняли из материала главы 2, количество инструментальных средств, а следовательно, возможности по творческой обработке изображений в Photoshop поистине безграничны. Но, несмотря на это оптимистическое заявление, следует помнить о том, что существует ряд задач, выполнение которых вручную проблематично либо сопряжено с немалыми затратами времени. Примером подобной задачи может служить изменение размера документа. Когда документ единственный и неповторимый, подобная задача легко выполняема вручную, но когда количество документов переваливает за сотню, вы начнете сожалеть о напрасно прожитых молодых годах. Но все (или почти все) можно улучшить, исправить, модернизировать, поэтому не является исключением и этот случай. Достаточно воспользоваться сценарием (написанным вами лично или загруженным из Internet).

Для написания сценариев, выполняемых в среде Photoshop CS, могут применяться несколько способов, которые вкратце мы сейчас и рассмотрим.

Например, вы можете создавать сценарии, основанные на элементарных “кирпичиках”, которые еще называются *действиями* (или *макросами*). Обратите внимание на то, что макросы управляют исключительно элементами пользовательского интерфейса, а именно, пунктами меню, клавишами быстрого доступа, диалоговыми окнами и т.д. Если же воспользоваться *сценариями* в их классическом виде, можно получить доступ к недокументированным возможностям программы, что позволит значительно разнообразить доступные пользователю возможности. Обратите также внимание на то, макросы не позволяют управлять потоком,

а сценарии обеспечивают подобную возможность. На практике *поток* означает то, что разработчик может реализовывать деревья перебора, позволяющие выбирать один вариант среди множества альтернатив. Поточковая обработка позволяет выполнять множество операций в автоматизированном режиме, благодаря чему устраняется рутинная и значительно упрощается обработка больших массивов изображений. Обратите также внимание на то, что в состав сценариев можно включать переменные, благодаря чему могут сохраняться заранее определенные пользователем значения — размер вновь создаваемых изображений и т.д. С помощью сценариев можно также одновременно использовать несколько программ для обработки изображений. Например, если пользоваться программой верстки QuarkXPress, можно написать сценарий, который в автоматическом режиме воспринимает порядок действий по обработке изображения (обрезка, вращение, изменение размеров, режима цвета и т.д.). После этого в рамках исходного сценария можно открыть изображение в окне Photoshop, выполнить некоторые операции по его обработке, сохранить результаты, затем импортировать в программу верстки.

При написании сценариев в Photoshop CS могут применяться самые различные языки сценариев. Если вы работаете на платформе Windows, можете воспользоваться языками написания сценариев JavaScript, Visual Basic, а также любыми другими языками, которые поддерживают COM-технологии (Perl, VBScript или Python). Приверженцы платформы Macintosh могут обратиться к языкам написания сценариев AppleScript или JavaScript.



Обратите внимание на то, что среди перечисленных языков написания сценариев к разряду универсальных (в плане поддерживаемой платформы) может быть отнесен лишь язык JavaScript. Недостатком этого языка является отсутствие поддержки со стороны многих распространенных приложений, в том числе и популярного и любимого многими пользователями QuarkXPress.

Даже если вы никогда не решитесь написать собственный сценарий, можете воспользоваться готовым сценарием, “позаимствованным” из Internet.

Как упоминалось ранее, гибкие и разнообразные возможности обеспечивают сценарии JavaScript. Для того чтобы воспользоваться

предоставляемыми ими возможностями, достаточно скопировать их в папку Program Files\Adobe\Photoshop CS\Presets\Scripts\. При этом Photoshop отображает данные файлы в подменю Scripts меню File. Если ваш сценарий отсутствует в данной папке, укажите Photoshop, где его можно найти, выбрав пункт Browse (Просмотр) в подменю Scripts (Сценарии).

Обратите внимание на то, что в комплект поставки Photoshop входят четыре стандартных сценария, указанных в следующем перечне.

- **Export Layers to Files** (Экспорт слоев в файлы). Назначение этого сценария заключается в том, чтобы экспортировать слои документа, в результате чего создается отдельный файл со сведенными слоями. При запуске этого сценария на выполнение пользователю предлагается выбрать формат сохраняемого файла.
- **Export Layer Comps to Files** (Экспорт композиций слоев в файлы). Если вы использовали палитру Layer Comps для создания композиций слоев, с помощью этого сценария можно сохранять подобные слои в виде файла со сведенными слоями. В данном случае пользователь может также выбрать формат файла.
- **Export Layer Comps to PDF** (Экспорт композиций слоев в файл PDF). При запуске этого сценария на выполнение производится сохранение композиций слоев в виде одного PDF-файла (причем каждый слой сохраняется в виде отдельной страницы). Этот сценарий весьма удобен для предварительного макетирования изображений.
- **Export Layer Comps to WPG** (Экспорт композиций слоев в галерею фотографий, размещенных в Web). Этот сценарий будет полезен в том случае, если вы собираетесь выгружать композиции слоев на Web-сайт. Подробнее тема размещения цифровых фотографий в Internet будет рассмотрена в главе 5.

Обратите внимание на текст сценария Export Layer Comps to WPG, который приводится в листинге 3.1.

Листинг 3.1. Сценарий Export Layer Comps to WPG

```
// (c) Copyright 2003. Adobe Systems, Incorporated.
// Все права защищены.
// Этот сценарий применяется к каждой композиции слоя,
// выполняя экспортирование в файл

// Уровень отладки: 0-2 (0:отключить, 1:останов при
// наличии ошибки, 2:точка прерывания в начале)
$.level = 0;
// запуск отладчика в следующей строке

//=====
// Глобальные переменные
//=====

// Строки интерфейса пользователя, которые
// могут быть русифицированы
var uiTitleLayerCompsToWPG = "Layer Comps To WPG";
var uiButtonRun = "Run";
var uiButtonCancel = "Cancel";
var uiHelpText = "Please specify the location where flat
  image files should be saved. Once Photoshop has saved
  these files, it will launch Web Photo Gallery in order
  to convert each file into a Web page.";
var uiLabelDestination = "Destination:";
var uiButtonBrowse = "Browse...";
var uiLabelStyle = "Style:";
var uiCheckboxSelectionOnly = "Selected Layer Comps Only";
var uiAlertSpecifyDestination = "Please specify destination.";
var uiAlertDestinationNotExist = "Destination does not
  exist.";
var uiTitleSelectDestination = "Select Destination";
var uiAlertDocumentMustBeOpened = "You must have a document
  open to export!";
var uiAlertNoLayerCompsFound = "No layer comps
  found in document!";
var uiAlertWasSuccessful = " was successful.";
var uiAlertFailed = " failed.";
var uiStyle = "Simple";

// Настройки
var clipSize = 5000; // изменение размера до вызова WPG
(пиксели)

//=====
// Функции
//=====

function settingDialog(exportInfo)
{
  var dlgW = 380; // ширина диалогового окна
```

```

var dlgH = 230; // высота диалогового окна
var btnW = 80; // ширина кнопки
var btnH = 20; // высота кнопки
var txtW = 260; // ширина текста
var txtH = 20; // высота текста
var mgn = 10; // граница

var bounds;
var yLayout = mgn+txtH; // сканирование схемы

bounds = {x:0, y:0, width:dlgW, height:dlgH};
var w = new Window("dialog", uiTitleLayerCompsToWPG,
    bounds );

bounds = {x:dlgW-btnW-mgn, y:yLayout,
    width:btnW, height:btnH};
w.btnRun = w.add("button", bounds, uiButtonRun);
w.btnRun.onClick = btnRunOnClick;

yLayout += btnH + mgn;
bounds = {x:dlgW-btnW-mgn, y:yLayout,
    width:btnW, height:btnH};
w.btnCancel = w.add("button", bounds, uiButtonCancel);
w.btnCancel.onClick = function() { this.parent.close(0);
};
yLayout = mgn; // переустановить

bounds = {x:mgn, y:yLayout, width:180, height:txtH};
// Если ширина ограничителя слишком велика, кнопка "просмотра"
// неэффективна.
w.add("statictext", bounds, uiLabelDestination);

yLayout += txtH;
bounds = {x:mgn, y:yLayout,
    width:txtW-btnW-mgn-5, height:txtH};
// уже на 5 пикселей
w.etDestination = w.add("edittext", bounds,
    exportInfo.destination.toString());

bounds = {x:mgn+txtW-btnW-5, y:yLayout,
    width:btnW+5, height:btnH}; // на 5 пикселей шире
w.btnBrowse = w.add("button", bounds, uiButtonBrowse);
w.btnBrowse.onClick = btnBrowseOnClick;

yLayout += txtH + mgn;
bounds = {x:mgn, y:yLayout, width:txtW, height:txtH};
w.add("statictext", bounds, uiLabelStyle);

yLayout += txtH;
bounds = {x:mgn, y:yLayout, width:txtW, height:txtH};
w.etStyle = w.add("edittext", bounds,
    exportInfo.style.toString());

```

```

yLayout += txtH + mgn;
bounds = {x:mgn, y:yLayout, width:txtW, height:txtH};
w.cbSelection = w.add( "checkbox", bounds,
    uiCheckboxSelectionOnly);
w.cbSelection.value = exportInfo.selectionOnly;

bounds = {left:mgn, top:dlgH-mgn-85, right:dlgW-mgn,
    bottom:dlgH-mgn};
w.pnlHelp = w.add( "panel", bounds, "" );
bounds = {x:mgn, y:mgn, width:dlgW-mgn*4, height:60};
w.pnlHelp.add("statictext", bounds, uiHelpText,
    {multiline:true});

w.center(); // центрирование диалогового окна
var result = w.show();
if ( 0 == result) return result;
// закрыть для выхода

// получение значения настройки из диалогового окна
exportInfo.destination = w.etDestination.text;
exportInfo.style = w.etStyle.text;
exportInfo.selectionOnly = w.cbSelection.value;

return result;
}

function btnRunOnClick()
{
    // проверка корректности установки
    var destination = this.parent.etDestination.text;
    if (destination.length == 0) {
        alert(uiAlertSpecifyDestination);
        return;
    }
    var testFolder = new Folder(destination);
    if (!testFolder.exists) {
        alert(uiAlertDestinationNotExist);
        return;
    }

    this.parent.close(1);
}

function btnBrowseOnClick()
{
    var defaultFolder = this.parent.etDestination.text;
    var testFolder = new
Folder(this.parent.etDestination.text);
    if (!testFolder.exists) defaultFolder = "~";
    var selFolder =
Folder.selectDialog(uiTitleSelectDestination, defaultFolder);
    if ( selFolder != null ) {
        this.parent.etDestination.text =

```

```

selFolder.toString();
    }
    return;
}

function initExportInfo(exportInfo)
{
    exportInfo.destination = new String("");
    exportInfo.tempLocation = new String("");
    exportInfo.style = new String(uiStyle);
    exportInfo.selectionOnly = false;

    try {
        exportInfo.destination =
            app.activeDocument.fullName.parent; //целевая папка
        var tmp = app.activeDocument.fullName.name;
        exportInfo.fileNamePrefix = decodeURI(tmp.substring(0,
            tmp.indexOf("."))); // часть имени файла
    } catch(someError) {
        exportInfo.destination = new String("");
        exportInfo.fileNamePrefix =
            app.activeDocument.name; // часть имени файла
    }
}

function setTempFolder(exportInfo)
{
    while(true) { // присвоение случайного имени
        // временной папке
        exportInfo.tempLocation = exportInfo.destination +
            "/temp" + Math.floor(Math.random()*10000);
        var testFolder = new Folder(exportInfo.tempLocation);
        if (!testFolder.exists) {
            testFolder.create();
            break;
        }
    }
}

function zeroSuppress (num, digit)
{
    var tmp = num.toString();
    while(tmp.length < digit) tmp = "0" + tmp;
    return tmp
}

function main()
{
    if ( app.documents.length <= 0 ) {
        alert( uiAlertDocumentMustBeOpened );
        return;
    }
}

```

```

var exportInfo = new Object();
initExportInfo(exportInfo);
if (0 == settingDialog(exportInfo)) return; // выход

try {
    var docName = app.activeDocument.name;
    exportInfo.fileNamePrefix.replace(/\\s/g, "_");
    // замена пробела знаком '_';

    setTempFolder(exportInfo);

    var compsName = new String("none");
    var compsCount = app.activeDocument.layerComps.length;
    if (compsCount <= 1) {
        alert ( uiAlertNoLayerCompsFound );
    } else {
        app.activeDocument = app.documents[docName];
        docRef = app.activeDocument;

    var tempFileList = new Array();

    var orgRulerUnits = app.preferences.rulerUnits;
    // сохранить единицу измерения
    app.preferences.rulerUnits = Units.PIXELS;

    var exportFileCount = 0;
    for (compsIndex = 0;
        compsIndex < compsCount; compsIndex++) {
        var compRef = docRef.layerComps[compsIndex];
        if (exportInfo.selectionOnly &&
            !compRef.selected) continue; // лишь выбрано
        compRef.apply();
        var duppedDocument = app.activeDocument.duplicate();
        if ((clipSize < duppedDocument.width.value)||
            (clipSize < duppedDocument.height.value)) {
            var wRatio = duppedDocument.width.value / clipSize;
            var hRatio = duppedDocument.height.value / clipSize;
            var ratio = Math.max(wRatio, hRatio);
            duppedDocument.resizeImage(duppedDocument.width/
                ratio, duppedDocument.height/ratio);
        }
        var fileNameBody = zeroSuppress(compsIndex+1, 2);
        // начало 01
        fileNameBody += "_" + compRef.name;
        fileNameBody = fileNameBody.replace(/\\/g, "_");
        // '/' -> '-'
        if (fileNameBody.length > 120)
            fileNameBody = fileNameBody.substring(0,120);
        var tempFile = exportInfo.tempLocation +
            "/" + fileNameBody + ".psd";
        tempFileList[exportFileCount] = tempFile;
        exportFileCount++;
        tempFile = new File( tempFile );
    }
}

```



```

        if (null != compRef.name)
            duppDocument.info.title = compRef.name;
        if (null != compRef.comment)
            duppDocument.info.caption = compRef.comment;
            duppDocument.saveAs( tempFile );
            duppDocument.close();
    }
    app.preferences.rulerUnits = orgRulerUnits;
    // восстановление единицы измерения

    // запуск Web Photo Gallery
    galleryOptions = new GalleryOptions();
    galleryOptions.layoutStyle = exportInfo.style;
    galleryOptions.addSizeAttributes = true;
    galleryOptions.preserveAllMetadata = true;
    galleryOptions.bannerOptions.siteName = docName;
    galleryOptions.imagesOptions.resizeImages = true;
    galleryOptions.imagesOptions.dimension = 450;
    galleryOptions.imagesOptions.includeFilename = true;
    galleryOptions.imagesOptions.caption = true;
    galleryOptions.imagesOptions.includeTitle = true;
    galleryOptions.thumbnailOptions.includeFilename =
true;
    galleryOptions.thumbnailOptions.caption = true;
    galleryOptions.thumbnailOptions.includeTitle = true;

    app.makePhotoGallery(Folder(exportInfo.tempLocation),
        Folder(exportInfo.destination), galleryOptions);

    // удаление временных файлов
    for ( compsIndex = 0; compsIndex < exportFileCount;
        compsIndex++ ) {
        tempFile = new File( tempFileList[compsIndex] );
        tempFile.remove();
    }
    // удаление временной папки
    var tempFolder = new Folder(exportInfo.tempLocation);
    tempFolder.remove();
    }
    } catch (e) {
        alert(e);
    }
}

//=====
// Отсылка
//=====

main();

```

Как видите, приведенный пример является достаточно сложным.



Если вы испытываете затруднения при чтении кода приведенного выше сценария, а также хотите сами разрабатывать нечто подобное, обратитесь к книге Дэнни Гудмена *JavaScript. Библия пользователя*, выпущенной издательством “Диалектика” в 2002 году. Впрочем, после чтения остальной части главы большая часть этого сценария станет понятной.

С ростом сложности сценария возрастает потребность в его “интерактивности” (т.е. более тесной “интеграции” с пользователем). Чем сложнее сценарий, тем больше он нуждается во взаимодействии с пользователем — с помощью кнопок, текстовых полей, диалоговых окон, всплывающих меню и т.д. Если вы будете заниматься разработкой пользовательских интерфейсов на основе сценариев AppleScript (естественно, на платформе Macintosh), воспользуйтесь приложением AppleScript Studio. На платформе Windows для этих же целей можно воспользоваться языками написания сценариев JavaScript и Visual Basic. Помимо упомянутой выше книги дополнительные сведения можно почерпнуть в руководстве JavaScript Reference Guide, которое входит в комплект поставки Photoshop CS.

Обратите внимание на то, что далеко не все операции могут выполняться в автоматизированном виде с помощью сценариев. Еще никому не удалось разработать сценарий, изменяющий масштаб изображения либо использующий фильтры (подключаемые модули), не входящие в состав стандартных фильтров Photoshop. Чтобы получить нестандартные возможности, лучше пользоваться JavaScript, хотя и он не всесилен. На самом деле можно воспользоваться “обходным маневром”, суть которого заключается в следующем. Просто загрузите подключаемый модуль ScriptingListener, который входит в состав набора файлов Scripting Support. Этот модуль генерирует код JavaScript, отражающий все операции, выполняемые в среде Photoshop CS. После этого можете воспользоваться сохраненным кодом для повторного автоматического выполнения этих действий.

К сценариям я еще вернусь, а пока прошу обратить ваши внимание на другие средства автоматизации в Photoshop CS, описанию которых посвящен следующий раздел.



Модуль ScriptingListener находится в каталоге \Program Files\Adobe\Photoshop CS\Scripting Guide\Utilites. Для инсталляции переместите его в каталог \Program Files\Adobe\Photoshop CS\ Plug-Ins\ Adobe Photoshop Only\Automate. Перезапустите Photoshop CS, после чего все выполняемые операции будут регистрироваться в файле C:\ScriptingListenerJS.log.

Макросы и меню Automate

Средства автоматизации, применяемые в Photoshop CS, можно разделить на две категории: меню Automate (Автоматизация) и макросы. Для начала подробнее опишем макросы.

Макросы

Как упоминалось ранее, подавляющая часть операций, выполняемых в Photoshop, трудно назвать творческой. Скорее, это регулярно повторяемая последовательность определенных действий. В этой ситуации напрашивалась идея “конвейера”, благодаря которому можно поручить выполнение рутинных операций компьютеру. И тут появились на сцене макросы (в английском варианте “actions”, иногда это слово переводится как “действия”). Сразу хочу отметить, что макросы не являются изобретением разработчиков из фирмы Adobe. Эти средства достаточно давно и плодотворно применяются фирмой Microsoft в своем офисном продукте Microsoft Office. По определению макрос — это записанная последовательность действий, выполнение которых необходимо для получения конкретного результата. Каждая команда, которая выполняется в среде Photoshop вручную, снабжена собственным уникальным именем, а также четко определенной областью действия. Например, одна из наиболее распространенных команд редактирования, которая находится в меню Edit (Редактирование), называется Cut (Вырезать), а ее область применения — существующая выделенная область.

В процессе использования макросов пользователю доступны три основные операции: создание, редактирование и выполнение. В этих целях применяется палитра Actions (Действия), которая чаще всего используется для работы с макросами (рис. 3.1).



Рис. 3.1. Вид палитры Actions в Photoshop CS

В состав палитры Actions входят следующие элементы.

- **Активный макрос.** В рассматриваемом случае стрелка указывает на макрос *Vignette*.
- **Набор макросов.** В комплект стандартной поставки Photoshop CS входит шесть папок, включающих наборы макросов:
 - **Default Actions (Макросы, заданные по умолчанию).** Набор разнообразных и наиболее часто применяемых макросов (по мнению разработчиков из Adobe);
 - **Commands (Команды).** В этот набор входят самые распространенные команды: *Copy* (Копировать), *Grow* (Смежные пиксели) и тому подобное (обратите внимание на то, что каждому макросу может быть назначена отдельная функциональная клавиша).
- **Toggle DiaLog on/off (Отобразить/скрыть диалоговое окно).** В полном соответствии со своим названием выполняет отображение или сокрытие диалогового окна.
- **Toggle Item on/off (Воспроизводить/не воспроизводить макрос).** Определяет воспроизведение макроса, выбранного в списке.
- **Stop Playing/Recording (Остановить воспроизведение/запись).** При щелчке на этой пиктограмме прекращается воспроизведение или запись макроса.
- **Begin Recording (Начать запись).** Начать запись макроса.
- **Play Selection (Выполнить).** Выполнить имеющийся макрос. Основная кнопка при работе с макросами. Но гораздо удобнее запускать макрос или сочетанием клавиш, которое вы можете закрепить за любым макросом сами, или двойным щелчком мыши при нажатой клавише <Ctrl>.

- **Create New Set (Создать набор).** После щелчка на этой пиктограмме создается новая группа макросов. Если макросы записываются пользователем, удобно щелчком на этой кнопке создать папку, в которую “сбросить” макросы, в противном случае вы можете попасть в такую ситуацию, когда будете долго и мучительно вспоминать, что же все-таки делает макрос `My_action`.
- **Create New Action (Создать макрос).** Тут все ясно, достаточно щелкнуть на этой кнопке, как будет создан новый макрос.
- **Delete (Удалить).** Эта пиктограмма имеет вид урны, поэтому ее назначение понятно без дополнительных комментариев.

Если щелкнуть на круглой пиктограмме со стрелкой в правом верхнем углу палитры, откроется меню палитры **Actions** (рис. 3.2). С помощью этого меню можно выполнить все команды, доступные в данный момент в палитре.



Рис. 3.2. Меню палитры Actions

Пример создания макроса

Давайте попробуем создать новый макрос. Его назначение заключается в преобразовании изображения в полутоновой режим, уменьшении первоначальных размеров в три раза, а также сохранении результатов изменений. Прежде чем приступить к этой работе, следует

создать новую папку, в которой будут храниться наши макросы. В процессе создания папки следует присвоить “говорящее имя”, облегчающее распознавание ее назначения в дальнейшем. В нашем случае можно назвать папку `First Macros` (Первый макрос). После этого щелкните на пиктограмме `Create New Action`. В результате откроется диалоговое окно `New Action` (Новый макрос), которое показано на рис. 3.3. В этом окне пользователю предлагается указать значения следующих параметров.



Рис. 3.3. Диалоговое окно `New Action`

- **Name (Имя).** Название этого поля говорит само за себя — здесь указывается имя нового макроса. В данном случае вполне логично выбрать название `Convert_to_grayscale`.
- **Set (Набор).** В этом поле определяется имя набора, в состав которого будет входить данный макрос.
- **Function Key (Функциональная клавиша).** Здесь определяется “горячая” клавиша, применяемая для запуска на выполнение данного макроса. Давайте выберем `<Ctrl+F12>`.
- **Color (Цвет кнопки).** Настраивать этот параметр стоит лишь в том случае, если для палитры `Actions` выбран режим `Button Mode` (Кнопочный режим). Переключение в этот режим происходит после выбора пункта `Button Mode` в меню палитры.

После щелчка на кнопке `Record` (Запись макроса) все действия, выполняемые пользователем, будут регистрироваться в макросе. Еще раз напомним о том, что функция создаваемого макроса будет заключаться в преобразовании изображения в полутоновой режим, уменьшении его размеров в три раза и в сохранении изменений с последующим закрытием файла.



Помните о том, что при записи макроса обязательно должен быть открыт файл. Если вы этого еще не сделали, остановите запись макроса, щелкнув на пиктограмме **Stop Recording**, затем откройте или создайте любой графический файл и щелкните на пиктограмме **Begin Recording**.

Ну а теперь переходим непосредственно к процедуре записи макроса. Сначала преобразуем изображение в полутоновой режим, выбрав команду **Image⇒Mode⇒Grayscale** (Изображение⇒Режим⇒Полутоновые).

Теперь уменьшаем размер изображения в три раза. Для этого выберите команду **Image⇒Image Size** (Изображение⇒Размер изображения). В отобразившемся диалоговом окне в качестве единиц измерения укажите проценты (percent), а в полях **Width** (Ширина) и **Height** (Высота) введите значение 33,3%. Обратите внимание на то, что если флажок **Resample Image** (Пересчитать изображение) не установлен, доступ к полям **Height** и **Width** будет заблокирован. Теперь осталось сохранить изображение (клавиши <Ctrl+S>) и закрыть его.

Щелкните на пиктограмме **Stop Recording** для прекращения процедуры записи макроса. Теперь достаточно открыть любое изображение, нажать клавиши <Ctrl+F12>, а все остальное компьютер сделает за вас. Как видите, при выполнении рутинных операций макросы просто незаменимы. Если вы решите, что подобные простейшие операции автоматизировать не стоит, просто удалите макрос, щелкнув на пиктограмме **Delete**.

Изменение созданных макросов

Ранее созданный макрос может в дальнейшем изменяться. Необходимость в этом возникает достаточно часто, и зачастую связана с тем, что большинство действий, записанных в макросе, имеют собственные параметры. А эти параметры изменять приходится достаточно часто, в зависимости от выполняемой задачи и особенностей обрабатываемого изображения. Обратите внимание на рис. 3.4, где показан раскрытый макрос, заданный по умолчанию (*vignette*). Если, например, потребуется изменить параметры действия **Fill**, щелкните дважды на нем мышью. Изменения в макросе записываются автоматически.



Рис. 3.4. Изменение параметров заданного по умолчанию макроса Vignette

Запуск макросов на выполнение

Для запуска макроса на выполнение можно воспользоваться одним из трех способов:

- перейдите в палитру Actions и щелкните на кнопке Play Selection;
- удерживайте клавишу <Ctrl> и щелкните мышью дважды на выбранном макросе;
- нажмите клавиши быстрого доступа, присвоенные данному макросу.

Рекомендуется воспользоваться третьим вариантом, поскольку именно в этом случае реально экономится время.

И теперь некоторые замечания относительно стандартных макросов, входящих в комплект поставки Photoshop CS. Многие пользователи задаются вопросом, стоит ли использовать эти средства. Поскольку они уже имеются, то можно и воспользоваться, если, конечно, отсутствует более достойная альтернатива. И особенно ценными они будут в том случае, если снабжены клавишами быстрого доступа, благодаря которым существенно ускоряется работа.

Теперь обратимся к обзору средств автоматизации, находящихся в меню Automate.

Возможности меню Automate

Все инструментальные средства, предназначенные для автоматизации работы в Photoshop CS (за исключением макросов), вынесены в отдельное меню, доступ к которому открывается в меню File (команда File⇒Automate (Файл⇒Автоматизировать)). Перейдем к подробному рассмотрению этих возможностей.

Команда Batch

После выполнения этой команды (File⇒Automate⇒Batch (Файл⇒Автоматизировать⇒Пакет)) открывается диалоговое окно Batch, показанное на рис. 3.5.



Рис. 3.5. Диалоговое окно команды Batch

Ниже описаны параметры, отображаемые в этом диалоговом окне.

- **Set (Набор).** Определение набора, из которого выбирается макрос.
- **Action (Макрос).** Здесь определяется выбираемый пользователем макрос.
- **Source (Источник изображений).** В этом поле определены четыре стандартных значения: **Folder** (выбор изображения из локальной папки), **Import** (импортирование изображения), **Opened Files** (уже открытые файлы), **File Browser** (файлы, открытые в окне обозревателя файлов).
- **Choose (Выбрать).** После щелчка на этой кнопке пользователь может указать путь к папке, где хранятся изображения.
- **Override Action "Open" Commands (Игнорировать команды макроса "Открыть").** Если установлен этот флажок, Pho-

toshop CS будет игнорировать собственную команду открытия файла, указанную в макросе, а воспользуется источником графических изображений, который был указан выше.

- **Include All Subfolders** (Вложенные папки). При установке этого флажка будут открываться изображения, находящиеся во всех подпапках исходной папки.
- **Suppress File Open Options Dialogs** (Не отображать параметры в окне открытия файлов). При установке этого флажка не отображаются параметры, определенные в диалоговом окне открытия файлов.
- **Suppress Color Profile Warnings** (Не показывать цветовые предупреждения). При установке этого флажка блокируется отображение информации о цветах, которые не попадают в область полиграфического цветового охвата.
- **Destination** (Назначение). В этом поле со списком определяются действия с исходными файлами, которые будут выполнены по завершении макроса. Пользователь может выбрать один из двух вариантов:
 - **Save and Close** (Сохранить и закрыть). Этот параметр определяет перезапись исходных изображений новыми;
 - **Folder** (Папка). Если требуется предохранить исходные изображения от изменений, определите папку, в которую они копируются в неизменном виде.
- **Choose** (Выбрать). После щелчка на этой кнопке можно указать путь к папке, где хранятся обработанные изображения.
- **Override Action "Save In" commands** (Игнорировать команду макроса "Сохранить в"). При установке этого флажка игнорируются команды **Save In** (Сохранить в), определенные в макросе.
- **File Naming** (Имена файлов). Данная группа параметров определяет имена, под которыми сохраняются файлы. Есть множество автоматизированных вариантов, например последовательная нумерация, добавление даты и т.д.
- **Compatibility** (Совместимость). При установке этого флажка можно назначать имена файлов, учитывая совместимость с операционными системами OS/2 и UNIX.

- **Errors (Обработка ошибок).** В этом поле определяются действия, выполняемые в случае возникновения ошибок. Пользователю доступны два варианта выбора:
 - **Log Errors to File (Регистрировать ошибки в файле).** При выборе этого параметра задается регистрация ошибок в файле с продолжением выполнения макроса;
 - **Stop For Errors (Останов при обнаружении ошибок).** При выборе этого параметра выполнение макроса останавливается в случае обнаружения ошибок.

Как видите, команда **Batch** является одной из самых полезных команд, применяемых для автоматизированной обработки группы файлов изображений. Используя ее, пользователь может применить любой макрос ко всем файлам, находящимся в одной папке. Можно, например, выполнить обработку огромного количества фотографий, “отшелканных” цифровым фотоаппаратом, при этом вы сэкономите колоссальное количество времени. Наиболее частыми задачами, выполняемыми при обработке фотографий, является изменение размеров и цветового режима фотографий. Именно для подобных случаев эта команда является оптимальным выбором.

Команда PDF Presentation

Эта команда (**File⇒Automate⇒PDF Presentation (Файл⇒Автоматизировать⇒Презентация в формате PDF)**) появилась в версии Photoshop CS. С ее помощью пользователь может создавать презентации в формате PDF или многостраничные документы PDF. Под презентацией в данном случае подразумевается стандартное слайд-шоу, которое хранится в формате PDF. После его запуска на выполнение картинки будут сменять друг друга через определенные промежутки времени. Этот инструмент может оказаться полезным в том случае, если требуется представить результаты “творческих усилий”, воспользовавшись для этого Internet. После выбора этой команды отображается диалоговое окно PDF Presentation, которое показано на рис. 3.6.



Рис. 3.6. Диалоговое окно команды PDF Presentation

Ниже описаны параметры, представленные в этом диалоговом окне.

- **Source File (Источник).** В этом разделе выбираются файлы, которые будут представлены в будущей презентации. Если щелкнуть на кнопке **Browse (Просмотр)**, можно найти требуемые файлы на жестком диске. При установленном флажке **Add Open File (Добавить открытые файлы)** в презентацию будут добавлены также изображения, открытые в данный момент в Photoshop.
- **Output Options (Настройки сохранения).** В этом разделе находятся параметры, определяющие тип создаваемого документа:
 - **Save as (Сохранить как).** Пользователь может выбрать один из следующих двух переключателей: **Multi-Page Document (Многостраничный документ)** и **Presentation (Презентация)**;
 - **View PDF After Saving (Открыть файл PDF после сохранения).** Если этот флажок установлен, то сразу после завершения экспорта будет открыто окно Acrobat Reader для просмотра созданного файла.

- **Presentation Options** (Настройки презентации). В этом разделе доступны следующие параметры:
 - **Advance Every () seconds** (Смена каждые () секунд). Здесь определяется частота смены слайдов в презентации;
 - **Loop After Last Page** (Повтор после последней страницы). Если этот флажок установлен, презентация не завершится на последнем кадре, а будет проигрываться заново;
 - **Transition** (Смена). Определяется способ смены слайдов. Пользователь может выбрать один из 18 предварительно заданных вариантов.

Команда **Create Droplet**

Эта команда (File⇒Automate⇒Create Droplet (Файл⇒Автоматизировать⇒Создать дроплет)) очень похожа на команду **Batch**, правда, в данном случае не требуется указывать источник изображений. Этого не требуется по той причине, что в результате выполнения команды создается исполняемый файл (с расширением `.exe`), который помещается в папку с обрабатываемыми изображениями и выполняет заданный макрос.

Команда **Conditional Mode Change**

С помощью этой команды (File⇒Automate⇒Conditional Mode Change (Файл⇒Автоматизировать⇒Изменить цветовой режим)) можно изменять цветовой режим для открытых изображений (диалоговое окно представлено на рис. 3.7). Очень сильно напоминает команду **Image⇒Mode** (Изображение⇒Режим).



*Рис. 3.7. Диалоговое окно команды **Conditional Mode Change***

Команда Contact Sheet II

Эта команда (File⇒Automate⇒Contact Sheet II (Файл⇒Автоматизировать⇒Индекс)) позволяет сформировать изображение, на котором будут показаны в миниатюре все картинки из папки-источника. Команда незаменима при необходимости создания небольших индексных изображений, например в полиграфии. На рис. 3.8 показано диалоговое окно этой команды. Ниже описаны параметры, представленные в этом диалоговом окне:

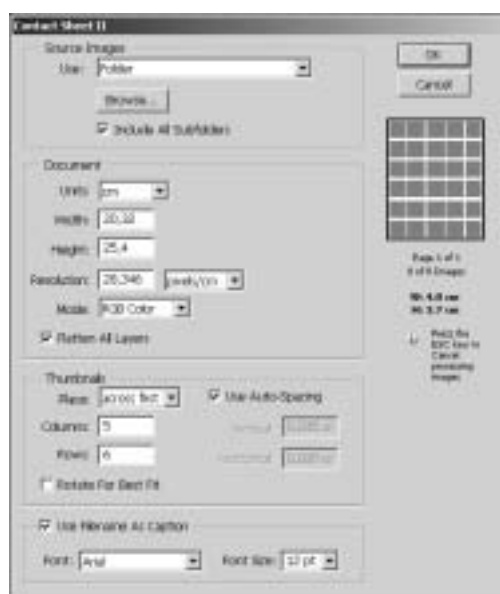


Рис. 3.8. Диалоговое окно Contact Sheet II

- Source Images (Исходные изображения). Здесь определяются исходные изображения, включаемые в индексы. Доступны следующие три варианта: Current Open Documents (Текущие открытые документы), Folder (Папка), Selected Images from File Browser (Изображения, выделенные в окне обозревателя файлов). После щелчка на кнопке Browse можно указать путь к графическим файлам. Флажок Include All Subfolders (Включить все подпапки) позволяет включить в обработку все подпапки с изображениями, находящиеся в данной папке.

- **Document (Документ).** Эта группа параметров определяет размеры документа, в котором будут размещены миниатюры изображений из папки-источника. Если все они не поместятся на один лист, программа создаст еще один или несколько листов, в зависимости от потребности в этом. Здесь же указывается разрешение, цветовой режим, а флажок **Flatten All Layers (Свести все слои)** позволяет свести все слои документа.
- **Thumbnails (Миниатюры).** Данная группа параметров позволяет определить способ размещения миниатюр.
- **Use Filename As Caption (Использовать имена файлов в качестве заголовков).** Этот флажок определяет использование имени файла в качестве подписи к изображению. Следует применять, если имя файла несет в себе ценную информацию. Здесь же определяется используемый шрифт и его размер.

Команда **Crop And Straighten Photos**

Эта команда (**File⇒Automate⇒Crop and Straighten Photos (Файл⇒Автоматизировать⇒Разрезать и разделить фотографии)**) позволяет быстро распознать и разделить разные фотографии, собранные в одном изображении. В какой-то мере действие этой команды противоположно действию предыдущей команды. Диалоговое окно отсутствует.

Команда **Fit Image**

Эта команда (**File⇒Automate⇒Fit Image (Файл⇒Автоматизировать⇒Подобрать размер)**) позволяет изменить размер изображения в соответствии с вашими предпочтениями (рис. 3.9). Размеры задаются в полях **Width (Ширина)** и **Height (Высота)**.



*Рис. 3.9. Диалоговое окно команды **Fit Image***

Команда **Multi-Page PDF to PSD**

Эта команда (**File⇒Automate⇒Multi-Page PDF to PSD (Файл⇒Автоматизировать⇒Преобразовать из PDF в PSD)**) позволяет преобразовывать многостраничные PDF-файлы в формат PSD,

который является “родным” для Photoshop CS. Команда весьма полезна, поскольку огромное количество графической информации, циркулирующей в Internet, хранится именно в формате PDF. На рис. 3.10 представлено диалоговое окно этой команды.

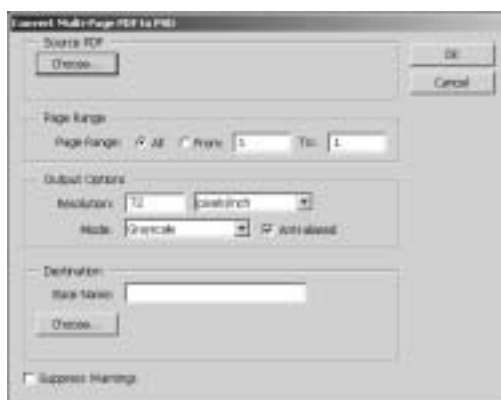


Рис. 3.10. Диалоговое окно команды Multi-Page PDF to PSD

Ниже описаны параметры, представленные в этом диалоговом окне.

- **Source PDF** (Исходный файл в формате PDF). После щелчка на кнопке Choose (Выбор) вы можете указать путь к обрабатываемому файлу.
- **Page Range** (Диапазон страниц). Эта группа параметров позволяет задать страницы, включенные в процесс преобразования.
- **Output Options** (Параметры сохранения). Здесь определяются параметры сохранения файлов в формате PSD.
- **Destination** (Назначение). Здесь определяется имя сохраняемого файла в формате PSD.

Команда Picture Package

Эта команда (File⇒Automate⇒Picture Package (Файл⇒Автоматизировать⇒Комплект страниц)) позволяет различным образом расположить на листе несколько копий изображения. В диалоговом окне команды (рис. 3.11) вниманию пользователя предлагается огромное количество готовых вариантов.



Рис. 3.11. Диалоговое окно команды *Picture Package*

В диалоговом окне команды вы найдете множество параметров, смысл которых очевиден, поэтому здесь объясняться не будет.

Команда Web Photo Gallery

Эта команда (File⇒Automate⇒Web Photo Gallery (Файл⇒Автоматизировать⇒Фотогалерея для Web)) позволяет подготовить изображения для размещения в Internet. Диалоговое окно этой команды представлено на рис. 3.12.

Параметры в этом диалоговом окне также не требуют особых пояснений.

Команда Photomerge

Эта команда (File⇒Automate⇒Photomerge (Файл⇒Автоматизировать⇒Склеивание фотографий)) появилась в версии Photoshop CS. Позволяет автоматизировать создание панорамных снимков на основе нескольких обычных фотографий. Диалоговое окно команды представлено на рис. 3.13.



Рис. 3.12. Команда Web Photo Gallery



Рис. 3.13. Диалоговое окно команды Photomerge



Если вы не являетесь специалистом в области обработки цифровых фотографий, лучше откажитесь от использования этой команды. Для достижения хороших результатов от вас потребуются нетривиальные знания.

На этом обзор команд, представленных в меню Automate, завершен. “Под занавес” поупражняемся в написании простого сценария на языке JavaScript.

Пример создания сценария

Создадим простой сценарий на языке JavaScript, который отображает на экране сообщение “Hello, World” (“Здравствуй, мир”). Для начала потребуется любой текстовый редактор, позволяющий сохранять результаты в виде простых текстовых файлов. Вполне подойдет Notepad (Блокнот), входящий в комплект поставки любой версии Windows.

После открытия диалогового окна редактора Notepad введите следующий сценарий на языке JavaScript. Сценарий снабжен комментариями, благодаря которым смысл выполняемых им действий становится очевидным.

Листинг 3.2. Поэкспериментируйте с этим простым сценарием

```
// Сценарий типа "Hello, Word!"
// Сохранение текущих единиц измерения, присвоение этим
// единицам значений, требуемых сценарию
var originalUnit = preferences.rulerUnits;
preferences.rulerUnits = Units.INCHES;

// Создание нового документа размером 4 на 4 дюйма и
// присвоение его переменной
var docRef = app.documents.add( 4, 4 );

// Создание нового слоя, содержащего текст
var artLayerRef = docRef.artLayers.add();
artLayerRef.kind = LayerKind.TEXT;

// Установка содержимого текстового слоя.
var textItemRef = artLayerRef.textItem;
textItemRef.contents = "Hello, World!";

// Обнуление ссылок
docRef = null;
artLayerRef = null;
textItemRef = null;

// Восстановление исходных настроек единиц измерения
app.preferences.rulerUnits = originalUnit;
```

Теперь сохраните этот сценарий под именем `first_script.js`, поместите его в папку `C:\Program Files\Adobe Photoshop CS\Presets\Scripts`. После перезапуска Photoshop CS этот сценарий станет доступным в меню `File⇒Scripts`. Выберите его и запустите на выполнение. Результат показан на рис. 3.14.

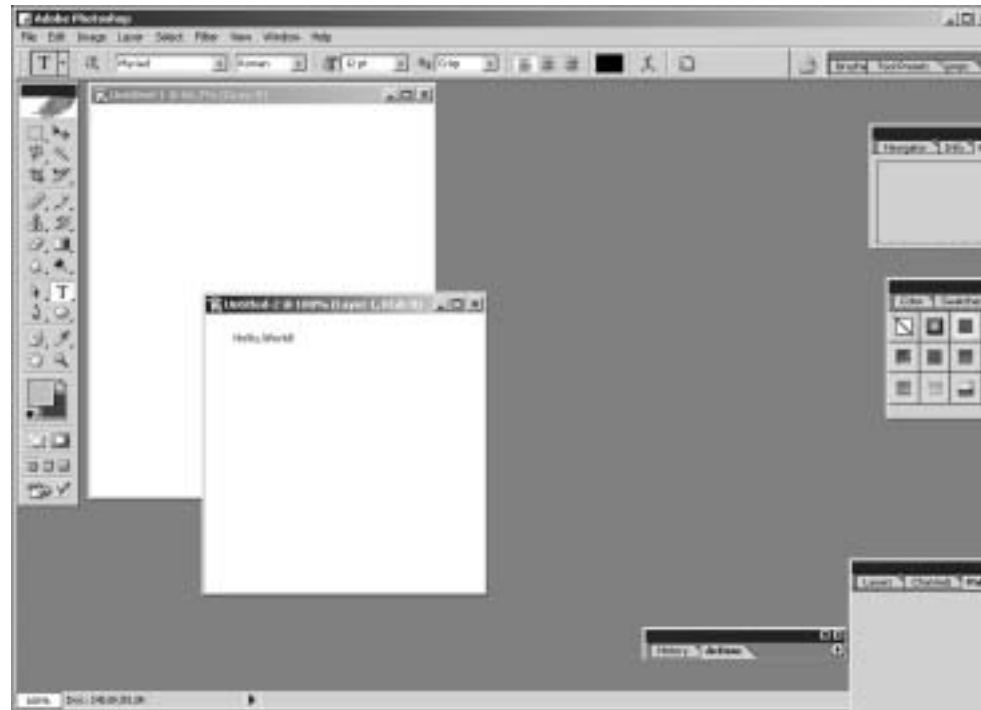


Рис. 3.14. Результат выполнения нашего первого сценария

Контрольные вопросы

1. Назовите различия между макросами и сценариями.
2. Попробуйте составить псевдокод, иллюстрирующий работу сценария из листинга 3.1.
3. Как называется модуль, регистрирующий операции, выполняемые пользователем в среде Photoshop CS?
4. Для чего нужна команда Contact Sheet II?
5. Какая команда используется для пакетной обработки изображений?
6. Опишите назначение параметров в диалоговом окне команды Picture Package.
7. Опишите назначение параметров в диалоговом окне команды Web Photo Gallery.