

GETTING STARTED WITH FLASH[®] LITE[™] 2.x

© 2007 Adobe Systems Incorporated. All rights reserved.

Getting Started with Flash® Lite™ 2.x

If this guide is distributed with software that includes an end user agreement, this guide, as well as the software described in it, is furnished under license and may be used or copied only in accordance with the terms of such license. Except as permitted by any such license, no part of this guide may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, recording, or otherwise, without the prior written permission of Adobe Systems Incorporated. Please note that the content in this guide is protected under copyright law even if it is not distributed with software that includes an end user license agreement.

The content of this guide is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Adobe Systems Incorporated. Adobe Systems Incorporated assumes no responsibility or liability for any errors or inaccuracies that may appear in the informational content contained in this guide.

Please remember that existing artwork or images that you may want to include in your project may be protected under copyright law. The unauthorized incorporation of such material into your new work could be a violation of the rights of the copyright owner. Please be sure to obtain any permission required from the copyright owner.

Any references to company names in sample templates are for demonstration purposes only and are not intended to refer to any actual organization.

Adobe, the Adobe logo, Flash Lite, and Flash are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Third-Party Information

This guide contains links to third-party websites that are not under the control of Adobe Systems Incorporated, and Adobe Systems Incorporated is not responsible for the content on any linked site. If you access a third-party website mentioned in this guide, then you do so at your own risk. Adobe Systems Incorporated provides these links only as a convenience, and the inclusion of the link does not imply that Adobe Systems Incorporated endorses or accepts any responsibility for the content on those third-party sites.



Sorenson™ Spark™ video compression and decompression technology licensed from Sorenson Media, Inc.

Fraunhofer-IIS/Thomson Multimedia: MPEG Layer-3 audio compression technology licensed by Fraunhofer IIS and Thomson Multimedia (<http://www.iis.fhg.de/amm/>).

Independent JPEG Group: This software is based in part on the work of the Independent JPEG Group.

Nellymoser, Inc.: Speech compression and decompression technology licensed by Nellymoser, Inc. (<http://www.nelly-moser.com>).

Opera® browser Copyright © 1995-2002 Opera Software ASA and its suppliers. All rights reserved.

Macromedia Flash 8 video is powered by On2 TrueMotion video technology. © 1992-2005 On2 Technologies, Inc. All Rights Reserved. <http://www.on2.com>.

Visual SourceSafe is a registered trademark or trademark of Microsoft Corporation in the United States and/or other countries.

Updated Information/Additional Third Party Code Information available at <http://www.adobe.com/go/thirdparty/>.

Adobe Systems Incorporated, 345 Park Avenue, San Jose, California 95110, USA.

Notice to U.S. Government End Users. The Software and Documentation are "Commercial Items," as that term is defined at 48 C.F.R. §2.101, consisting of "Commercial Computer Software" and "Commercial Computer Software Documentation," as such terms are used in 48 C.F.R. §12.212 or 48 C.F.R. §227.7202, as applicable. Consistent with 48 C.F.R. §12.212 or 48 C.F.R. §§227.7202-1 through 227.7202-4, as applicable, the Commercial Computer Software and Commercial Computer Software Documentation are being licensed to U.S. Government end users (a) only as Commercial Items and (b) with only those rights as are granted to all other end users pursuant to the terms and conditions herein. Unpublished-rights reserved under the copyright

laws of the United States. Adobe Systems Incorporated, 345 Park Avenue, San Jose, CA 95110-2704, USA. For U.S. Government End Users, Adobe agrees to comply with all applicable equal opportunity laws including, if appropriate, the provisions of Executive Order 11246, as amended, Section 402 of the Vietnam Era Veterans Readjustment Assistance Act of 1974 (38 USC 4212), and Section 503 of the Rehabilitation Act of 1973, as amended, and the regulations at 41 CFR Parts 60-1 through 60-60, 60-250, and 60-741. The affirmative action clause and regulations contained in the preceding sentence shall be incorporated by reference.

Contents

Introduction	7
Guide to instructional media.....	7
Additional resources	8
Typographical conventions	8
Chapter 1: Flash Lite Overview	9
Hello World Flash Lite application	9
The Adobe Device Central emulator and device database.....	11
Workflow for authoring Flash Lite applications.....	11
About Flash Lite content types	13
Creating a Flash Lite document template.....	14
Chapter 2: Tutorial: Creating a Flash Lite Application	17
Cafe application overview.....	17
View the completed application	20
Create the application	22
Index	39

Introduction

This manual describes how to develop applications for Macromedia® Flash® Lite™ 2.0 and 2.1 from Adobe (collectively called Flash Lite 2.x) and how to test your content using the Adobe® Device Central CS3 emulator, which is part of Adobe® Flash® CS3 Professional. The primary difference between using Flash Lite in Flash CS3 and in previous versions of Flash is that the Flash Lite emulator is now part of Device Central. See the Device Central documentation for more information.

This chapter contains the following sections:

Guide to instructional media	7
Additional resources	8
Typographical conventions	8

Guide to instructional media

The Flash Lite documentation package includes the following media to help you learn how to create Flash Lite applications:

- *Getting Started with Flash Lite 2.x* provides an overview of Flash Lite technology and developing Flash Lite content for mobile devices. It also includes a step-by-step tutorial for creating a Flash Lite application.
- *Developing Flash Lite 2.x Applications* is a comprehensive guide to creating Flash Lite content and includes instructions for testing your applications in the Adobe Device Central emulator.
- *Flash Lite 2.x ActionScript Language Reference* describes all the ActionScript language features available to Flash Lite developers and provides example code.

- *Introduction to Flash Lite 2.x ActionScript* complements the language reference and lists the differences between ActionScript for Flash Lite 2.0 and the version of ActionScript in Flash Player 7, upon which Flash Lite 2.0 and 2.1 are based.
- The Flash Lite sample applications at www.adobe.com/go/learn_ft_samples_and_tutorials demonstrate key concepts and best practices discussed or mentioned in the written documentation.

Additional resources

For the latest information on developing Flash Lite applications, plus advice from expert users, advanced topics, examples, tips, and other updates, see the Mobile and Devices Developer Center at www.adobe.com/go/developer_flashlite.

For TechNotes, documentation updates, and links to additional resources in the Flash Lite developer community, see the Adobe Flash Lite Support Center at www.adobe.com/go/support_flashlite.

Typographical conventions

The following typographical conventions are used in this manual:

- *Italic font* indicates a value that should be replaced (for example, in a folder path).
- `Code font` indicates ActionScript code.
- *Code font italic* indicates an ActionScript parameter.
- **Bold font** indicates a verbatim entry.
Double quotation marks ("") in code examples indicate delimited strings. However, programmers can also use single quotation marks (').

This chapter contains the following topics:

Hello World Flash Lite application	9
The Adobe Device Central emulator and device database.	11
Workflow for authoring Flash Lite applications.	11
About Flash Lite content types.	13
Creating a Flash Lite document template.	14

Hello World Flash Lite application

This simple tutorial introduces you to the mobile authoring and testing features in Adobe® Flash® CS3 Professional, as well as the general workflow for creating content using Macromedia Flash Lite 2.x from Adobe. In this section, you create a simple Flash Lite application and test it in the Adobe Device Central CS3 emulator. For a more complete sample application, see [Chapter 2, “Tutorial: Creating a Flash Lite Application,” on page 17](#).

For the purposes of this tutorial, assume that you’re developing content for the Flash Lite stand-alone player. The procedure targets a generic device.

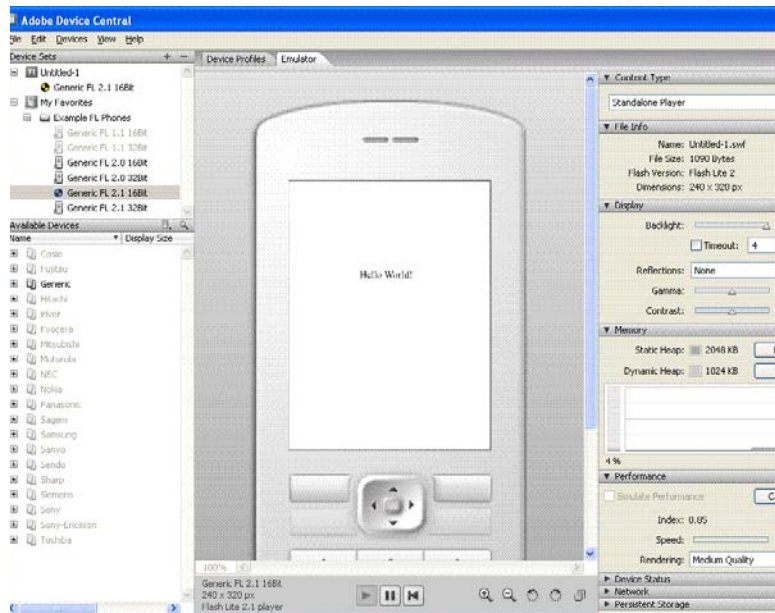
First, decide which devices and Flash Lite content type you are targeting.

To configure and create a simple Flash Lite application:

1. Start Flash.
2. On the main Flash screen, select **Create New > Flash Mobile Document**. Flash opens Adobe Device Central and displays the **New Document** tab.
3. In Device Central, select **FlashLite 2.0** in the **Player Version** box, **ActionScript 2.0** in the **ActionScript Version** box, and **Standalone Player** in the **Content Type** box.

4. Click Custom Size for All Selected Devices at the bottom of the screen. This allows you to create content for the stand-alone Flash Lite player.
5. Click Create. You are returned to Flash, which creates a new document with preset publish settings and (when you specify a device) the correct size for the device you selected.
6. In the Tools panel, select the Text tool and drag to create a text box on the Stage.
Type **Hello, world!** (or other text) in the text box.
7. Select Control > Test Movie to export your application to Adobe Device Central and view your application in the Adobe Device Central emulator.

Note: During testing in Device Central, you can change the device and content type to see your application on a different platform. To do this, double-click a device in the Available Devices panel and select a new content type from Content Type. When you return to Flash, Flash remembers the settings you last used in the emulator.



8. To return to Flash, select File > Return to Flash.

The Adobe Device Central emulator and device database

The Adobe Device Central emulator lets you test your content as it will run and appear on an actual device. You can select a different test device or content type and view detailed information about your application.

Device Central contains an extensive database of mobile devices from which you can select to create your own test environment. See the Device Central documentation for more information.

Workflow for authoring Flash Lite applications

Creating Flash Lite content is an iterative process that involves the following steps:

Identify your target devices and Flash Lite content type Different devices have different screen sizes, support different audio formats, and have different screen color depths, among other factors. These factors may influence your application's design or implementation.

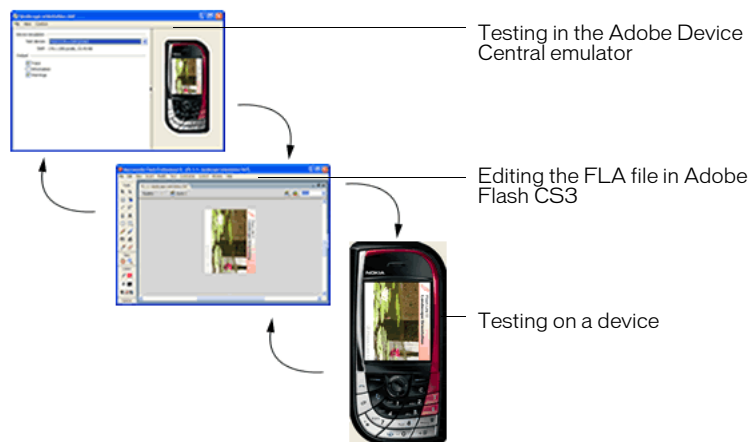
In addition, different devices support different Flash Lite content types, such as screen savers, stand-alone applications, or animated ring tones. The content type for which you are developing also determines the features that are available to your application. For more information, see [“About Flash Lite content types”](#) on page 13.

Create your application in Flash and test in Adobe Device Central

Central Adobe Flash CS3 Professional includes an emulator on Adobe Device Central that lets you test your application without having to transfer it to a device. You use the emulator to refine your application design and fix any problems before you test it on a mobile device.

Test the application on your target device or devices This step is important because the Adobe Device Central emulator doesn't emulate all aspects of the target device, such as its processor speed, color depth, or network latency. For example, an animation that runs smoothly on the emulator might not run as quickly on the device because of its slower processor speed, or a color gradient that appears smooth in the emulator may appear banded when viewed on the actual device. After you test your application on a device, you may find that you need to refine the application's design in the Flash authoring tool.

The following illustration shows the iterative development and testing process:



About Flash Lite content types

Before you start developing a Flash Lite application, you need to know the following:

The device or devices on which the content will run (*target devices*).

The Flash Lite player is installed on a variety of devices. For a list of devices that have Flash Lite installed, see the Supported Devices page on the Adobe website at www.adobe.com/go/mobile_supported_devices/.

The Flash Lite content types that the target devices support. Each Flash Lite installation supports one or more application modes (*content types*). For example, some devices use Flash Lite to enable Flash-based screen savers or animated ring tones. Other devices use Flash Lite to render Flash content that is embedded in mobile web pages. Not all content types support all Flash Lite features.

Each Flash Lite content type, paired with a specific device, defines a specific set of Flash Lite features that are available to your application. For example, a Flash application that is running as a screensaver is not typically allowed to make network connections or download data.

The Flash Lite testing features in Adobe Device Central let you test against multiple devices and different Flash Lite content types. This ability lets you determine if your application uses features that aren't available for the type of content that you are developing. For more information about selecting target devices and content types, see "Testing Flash Lite Content" in *Developing Flash Lite 2.x Applications*.

Creating a Flash Lite document template

When you're developing Flash Lite content, you'll often target the same device, or group of devices, and Flash Lite content type. For example, you might be developing content for the stand-alone Flash Lite player running on the Series 60 devices from Nokia. To facilitate the authoring process, it can be useful to create a template that has the appropriate Stage size, publish settings, test devices, and Flash Lite content type preconfigured for your application.

In this section, you'll create a document template that has a Stage size of 176 x 208 and targets the stand-alone Flash Lite 2.0 player. You can use this template to create new documents.

In Flash, select File > New.

On the General tab in the New Document dialog box, select Flash Mobile Document, and click OK.

Select File > Publish Settings to open the Publish Settings dialog box.

Click the Flash tab in the Publish Settings dialog box, select Flash Lite 2.0 from the Version pop-up menu, and then select ActionScript 2.0 from the ActionScript Version pop-up menu. Click OK.

Select Modify > Document to open the Document Properties dialog box.

Type **176** in the Width text box and **208** in the Height text box. Click OK.

If desired, enter any code you will typically include for the target device and content type. For example, the following steps add ActionScript to set the content to full screen:

Create a new layer by clicking the Insert Layer button at the bottom of the Timeline or selecting Insert > Timeline > Layer.

Select the layer name and type **ActionScript**.

Open the Actions panel (Window > Actions) and enter the following code:

```
fscommand2("FullScreen", true);
```

Select Control > Test Movie to open Device Central.

9. Click the Device Profiles tab.
10. In the list of Available Devices, expand the folders to select the desired device(s).
11. Select the desired device and drag it to the desired device set in the Device Sets panel.
12. Select File > Return to Flash to close Device Central and return to your Flash file.
13. Select File > Save As Template to save the document as a template.
14. In the Save As Template dialog box, type **Flash Lite 2-0 - Symbian Series 60** in the Name text box.
15. From the Category list, select **Global Phones**.
16. If desired, enter a description of the template in the Description text box (as many as 255 characters).

The description appears when the template is selected in the New Document dialog box.

17. Click Save.

After you create and save the template, you can use it to quickly create a new document that has the same publish settings, device settings, and Stage size.

To create a new document from the template you just created:

1. In Flash, select File > New.
2. In the New Document dialog box, select the Templates tab.
3. From the list of available template categories on the left, select Global Phones, which is the name of the category that you specified when you created the template.
4. From the list of templates on the right, select Flash Lite 2-0 - Symbian Series 60, which is the name of the template that you specified when you created the template.
5. Click OK to create the new document.

A new document that is configured with the same publish settings, device settings, and Stage size as the template appears.

Tutorial: Creating a Flash Lite Application

In this tutorial, you'll develop an Adobe Flash Lite application that promotes a fictional restaurant called Café Townsend. Users can view a list of specials at the restaurant, watch a video about the chef, and call the restaurant to make reservations. Users also set their preferred location for making reservations using a shared object, which stores their preference between application sessions.

This chapter contains the following topics:

Cafe application overview	17
View the completed application	20
Create the application	22

Cafe application overview

The application consists of three screens. The home screen that appears when the application starts contains some introductory text about the restaurant and a menu with three items: Specials, View Video, and Reservations.

To select a menu item, the user presses the device's up and down navigation keys to give the desired menu item focus, and then presses the select key to confirm the selection. On this screen, the user can also exit the application by pressing the right soft key, or set the preferred location for making reservations by pressing the left soft key.



The home screen

On the specials screen, the user can navigate a list of descriptions and images of the day's lunch specials. To navigate between specials, the user presses the device's right soft key (labeled Next). To return to the application's home screen, the user presses the left soft key (labeled Home).



The specials screen

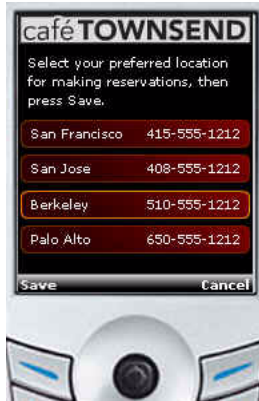
To view a short video about the café, the user selects the View Video menu option on the home screen. The user uses the left soft key to pause and play the video, as well as restart it when the video has finished playing.



Video screen

If the user selects the Reservations menu option on the home screen, the application initiates a phone call to the restaurant. By default, it dials the flagship restaurant in San Francisco, but the user can also specify another restaurant location to call by pressing the left soft key (Set Location) located on the application's home screen.

On the set location screen, the user can select the restaurant to set as the default location for making reservations. The location information is saved to a shared object, which persists between application sessions. The user can press the left soft key (Save) to save the new location setting, or press the right soft key (Cancel).



Screen to set location preference

View the completed application

A completed version of the café application is installed with Flash. You can view the completed application in the Adobe Device Central emulator, or if you have the stand-alone Flash Lite player installed on a mobile device, you can transfer the SWF file to your device to view it there.

1. In Flash, open the file named `café_tutorial_complete.fla` located at www.adobe.com/go/learn_ft_samples_and_tutorials. On the Samples and Tutorials page, locate, download and decompress the .zip file for your Flash Lite version, and then navigate to the Tutorial Assets folder to access the file.
2. Select Control > Test Movie to export the application to Device Central and run it in the Device Central emulator.
3. To interact with the application, do the following:
 - On the home screen, click the down key on the emulator's keypad to select the Specials menu item. Then click the select key on the emulator to view the specials screen.

On the specials screen, click the right soft key (Next) on the emulator to view the image and description for each special. Click the left soft key (Home) to return to the home screen.

- Select the View Video menu item to watch the video. Click the left soft key (Home) to return to the home screen. You can also click the right soft key (Replay) to view the video again.
- On the home screen, click the left soft key (Set Location) to go to the screen where you set your location. Select a preferred location for reservations and then click the left soft key (Save) to set the location and return to the home screen (or click Cancel).
- On the home screen, select the Reservations menu item to start a phone call to the restaurant. Confirm that the phone number the emulator displays matches the location you set on the set location screen.

Create the application

This section contains step-by-step procedures that show you how to re-create the café application. The tutorial is divided into the following parts:

- Creating the menu for the application’s home screen. From this screen, the user can select from a simple menu to view images and descriptions of the day’s specials, view a video, or to call the restaurant to make a reservation. The user can also press the device’s left soft key to go to the Options screen and set a preferred restaurant location. (See [“Create the menu for the home screen” on page 23.](#))
- Creating the specials screen. On this screen, users can press the device’s right soft key to navigate between images and descriptions for each lunch special at the café, or press the left soft key to return to the home screen. (See [“Create the specials screen” on page 27.](#))
- Creating the video screen. A video plays when this screen loads. Users can press the device’s left soft key to return to the home screen, or press the right soft key to restart the video. (See [“Create the video screen” on page 34.](#))
- Creating the set location screen. On this screen, users can select a preferred location to call for reservations. Pressing the device’s left soft key saves the selection and returns the user to the home screen, while pressing the right soft key returns them to the home screen without saving the location setting. (See [“Create the set location screen” on page 38.](#))

Selecting test devices and content type

You use Device Central to select the devices and content type that you are targeting. When you test your application in the Adobe Device Central emulator, the emulator configures itself to match the configuration of the player on the target device as well as the content type.

You specify these settings when you first create your Flash mobile document. For details on creating a new document from scratch, see “[Hello World Flash Lite application](#)” on page 9.

Create the menu for the home screen

In this section, you’ll create the menu for the application’s home screen. The menu consists of three options: Specials, View Video, and Reservations.

Open the partially completed source file named `cafe_tutorial_start fla` located at www.adobe.com/go/learn_ft_samples_and_tutorials. On the Samples and Tutorials page, locate, download and decompress the .zip file for your Flash Lite version, and then navigate to the Tutorial Assets folder to access the file.

In the Timeline window (Window > Timeline), select Frame 1 in the menu layer.

To create the menu, open the Library panel (Window > Library), and drag an instance of the button symbol called `specials button` to the Stage.

Position the button beneath the text field (already in place) that introduces the restaurant.

With the `specials button` selected, in the Property inspector, type `specials_btn` in the Instance Name text box.

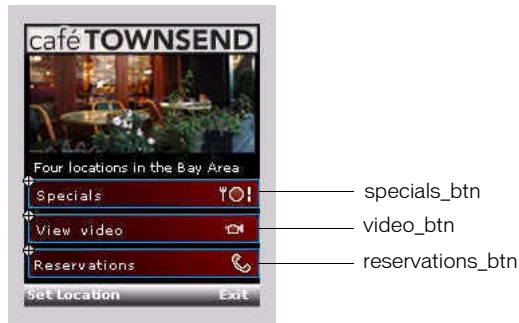
Drag an instance of the button symbol named `video button` to the Stage and position it below the `specials button`.

With the `video button` selected, in the Property inspector, type `video_btn` in the Instance Name text box.

Drag an instance of the button symbol named `reservations button` to the Stage and position it below the `video button`.

With the reservations button selected, in the Property inspector, type **reservations_btn** in the Instance Name text box.

The Stage of your application should look something like the following example:



In the Timeline, select Frame 1 in the layer named **ActionScript**.

Open the Actions panel (**Window > Actions**) and enter the following code:

```
stop();
_focusrect = false;
fscommand2("SetSoftKeys", "Set Location", "Exit");
fscommand2("SetQuality", "high");
fscommand2("Fullscreen", "true");
```

This code does the following:

- Stops the playhead at this frame.
- Disables the focus rectangle that Flash Lite draws by default around the button or input text field with the current focus (see “About the focus rectangle” in *Developing Flash Lite 2.x Applications*).
- Registers the soft keys for your application to use.
- Sets the player’s rendering quality to high. By default, Flash Lite renders graphical content at medium quality.
- Forces the player to display the application full screen.

When Flash Lite is in full-screen mode, the labels you specify in the `SetSoftKeys` command are not visible. For this reason, you must add custom soft-key labels to the Stage.

Add the following code to handle button events for the menu buttons, and for selection focus:

```
// Set initial focus when the application
// starts and also upon returning to the main
// screen from another screen.
if (selectedItem == null) {
    Selection.setFocus (specials_btn);
} else {
    Selection.setFocus (selectedItem);
}
// Assign onPress event handlers to each menu button,
// and set selectedItem variable to selected button
// object:
specials_btn.onPress = function () {
    gotoAndStop ("specials");
    selectedItem = this;
};
video_btn.onPress = function () {
    gotoAndStop ("video");
    selectedItem = this;
};
reservations_btn.onPress = function () {
    if (location_so.data.phoneNumber == undefined) {
        // User hasn't specified location so
        // go to "set location" screen:
        gotoAndStop ("options");
    }
    else {
        // Call number in shared object:
        var phoneNum = location_so.data.phoneNumber;
        getURL ("tel:" + phoneNum);
    }
    selectedItem = this;
};
```

The `onPress` event handlers assigned to the buttons named `specials_btn` and `video_btn` send the playhead to frames labeled, respectively, “specials” and “video.” You’ll create the content for those sections later in the tutorial (see [“Create the specials screen” on page 27](#) and [“Create the video screen” on page 34](#)).

When the user selects the Reservations option, the `onPress` handler dials the phone number specified in the `location_so` shared object. (Later in this procedure, you’ll create code to create the shared object.) If the user hasn’t yet specified a location to call for reservations, the application sends the playhead to the frame labeled “options,” where the user selects the preferred location for making reservations.

Now add the following code to create a key listener for the left and right soft keys:

```
Key.removeListener(myListener);
var myListener:Object = new Object();
myListener.onKeyDown = function() {
    var keyCode = Key.getCode();
    if (keyCode == ExtendedKey.SOFT1) {
        // Handle left soft key event:
        gotoAndStop("options");
    } else if (keyCode == ExtendedKey.SOFT2) {
        // Handle right soft key event:
        fscommand2("Quit");
    }
};
Key.addListener(myListener);
```

This code uses a key listener object to handle right and left soft key events. When the user presses the left soft key, the playhead is sent to the frame labeled “options,” and the right soft key closes the application.

For more information about using event listeners, see “Using a key listener to handle keypress events” in *Developing Flash Lite 2.x Applications*.

Finally, add code to initialize the shared object that saves the user’s preferred location for making reservations:

```
// Define Shared Object listener function:
function so_listener (the_so:SharedObject) {
    if (the_so.getSize () == 0) {
        // The shared object doesn't exist, so the user
        // hasn't set a preference yet.
    }
    SharedObject.removeListener ("location");
}
// Create shared object:
location_so = SharedObject.getLocal ("location");
// Add SharedObject listener object:
SharedObject.addListener ("location", this,
    "so_listener");
```

To test your work so far, select Control > Test Movie.

At this point you should be able to select a menu item by giving the corresponding button focus, and then pressing the emulator’s select key (or the Enter key on your computer keyboard). In the following sections, you’ll create the specials and video screens, as well as the screen to specify the default location.

Create the specials screen

In this section, you'll create the user interface elements that let the user browse images and descriptions of each special. The specials screen consists of the following parts:

- An animation that transitions between images of each special. (See [“Create the image animation” on page 27.](#))

- Dynamic text fields that display the name and description for each special. (See [“Add navigation and text to the specials screen” on page 31.](#))

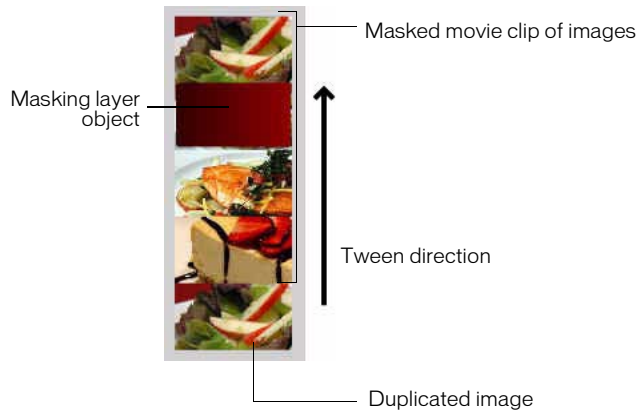
- User interface elements that let the user navigate between specials and return to the main application screen. (See [“Add navigation and text to the specials screen” on page 31.](#))

This section of the tutorial is divided into two parts. In the first part, you'll create the animation that transitions between images of each special. In the second part, you'll add user interface elements and ActionScript to let the user navigate between the images and to display each special's name and description.

Create the image animation

In this section, you'll create the tweened animation that transitions between images of each special. When you've completed this section, the animation plays through without stopping. Later in the tutorial, you'll add navigation and ActionScript that lets the user control the animation with the device's right soft key.

To create the animation, you'll use a prebuilt movie clip that contains images of all of the specials arranged in a vertical column. You'll use a masking layer to make only one of the images visible. Then you'll create a series of tweens that move the movie clip upward, so that a different image is visible. The last image in the movie clip is a duplicate of the first one, so that the animation sequence can return to its initial state after the user views the final image. These concepts are illustrated in the following example:



In the last section of the tutorial, you'll add ActionScript and user interface elements that let the user control the animation sequence.

Open the file you saved in [“Create the menu for the home screen” on page 23](#).

In the Timeline, select the keyframe in Frame 10 in the layer named Images.

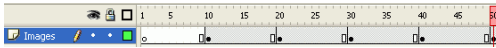
Open the Library panel, and drag the symbol named Specials movie clip to the Stage.

The rest of this tutorial refers to this movie clip simply as the *Images movie clip*.

With the new movie clip instance selected, set the movie clip's *x* and *y* coordinates both to **0** in the Property inspector.

This aligns the upper-left corner of the images movie clip with the upper-left corner of the Stage.

In the Images layer, insert keyframes in Frames 20, 30, 40, and 50, as the following image shows:



In the Timeline, select the keyframe in Frame 20.

On the Stage, select the Images movie clip, and set its *y* coordinate to **-100** in the Property inspector.

This moves the movie clip upward on the Stage 100 pixels.

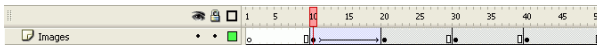
Select the keyframe in Frame 30 in the Timeline, select the images movie clip, and set its *y* coordinate to **-200** in the Property inspector.

Select the keyframe in Frame 40, select the images movie clip, and set its *y* coordinate to **-300** in the Property inspector.

Select the keyframe in Frame 50, select the images movie clip, and set its *y* coordinate to **-400** in the Property inspector.

Select the keyframe in Frame 10, and select Motion from the Tween pop-up menu in the Property inspector.

This tweens the Images movie clip's position between the keyframes in Frames 10 and 20.



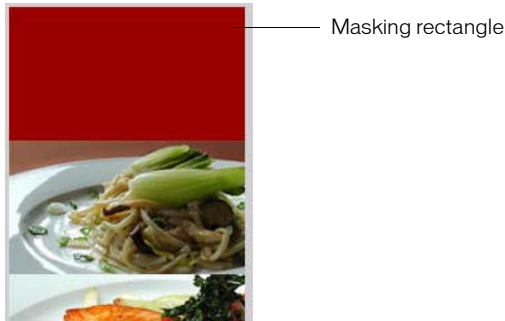
To create transitions between the other images, repeat step 11 for the keyframes located in Frames 20, 30, and 40.

To create the mask layer, select the Images layer in the Timeline, and then select Insert > Timeline > Layer (or click Insert Layer in the Timeline).

Insert a keyframe in Frame 10 of the new mask layer.

Using the Rectangle tool in the Tools palette, create a rectangle over the first (uppermost) image in the Images movie clip.

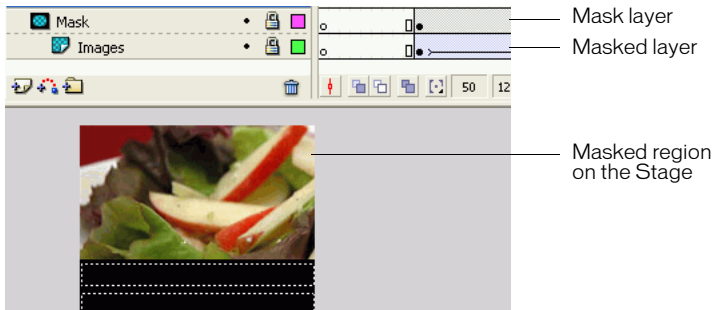
It doesn't matter what fill color you use for the rectangle, but it must be completely opaque.



To make sure the rectangle covers the entire image area, double-click the rectangle to select it, and then use the Property inspector to set its x and y coordinates both to **0**, its width to **176**, and its height to **100**.

Right-click (Windows) or Control-click (Macintosh) the Image Mask layer in the Timeline, and select Mask from the context menu.

The layer is converted to a mask layer, indicated by a mask layer icon. The layer immediately below it is linked to the mask layer, and its contents show through the filled area on the mask. For more information about working with mask layers in Flash, see "Using mask layers" in *Using Flash*.



Save your changes (File > Save).

At this point, if you were to test the application in the emulator, the animation you created would play through to the end and then stop. In [“Add navigation and text to the specials screen” on page 31](#), you’ll add ActionScript that stops the animation at each keyframe, as well as user interface elements that let the user navigate between images.

Add navigation and text to the specials screen

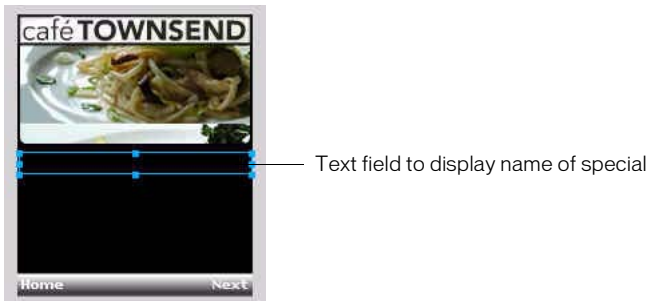
In this section, you’ll add interactivity to the specials screen that lets the user control the transition between each animation. You’ll also add dynamic text fields that display the name and description of each image.

In Flash, open the file you completed in [“Create the menu for the home screen” on page 23](#).

In the Timeline, select Frame 10 in the Text layer.

In the Tools palette, select the Text tool and create a text field below the first masked-specials image.

This text field displays the name of the special whose image is shown on the screen.

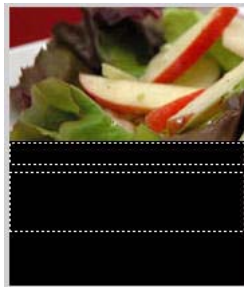


With the text field selected on the Stage, make the following changes in the Property inspector:

- Select Dynamic Text from the Text Type pop-up menu.
- Type `title_txt` in the Instance Name text box.
- Select the Italics text style option.
- Set the font size to 10.
- Select Use Device Fonts from the Font Rendering Method pop-up menu.

Create another text field below the first text field to display a short description of the specials that the user is viewing.

Using the Selection tool, resize the new text field so that it's about three times as tall as the text field above it.



Text field to display description of special

With the text field selected on the Stage, make the following changes in the Property inspector:

- Select Dynamic Text from the Text Type pop-up menu.
- Type **description_txt** in the Instance Name text box.
- Select Multiline from the Line Type pop-up menu.
- Set the font size to 10.
- Select Use Device Fonts from the Font Rendering Method pop-up menu.

In the Timeline, select the keyframe in Frame 10 in the ActionScript layer.

Open the Actions panel and add the following code:

```
stop();
fscommand2("SetSoftKeys", "Home", "Next");
title_txt.text = "Summer salad";
description_txt.text = "Butter lettuce with apples, blood
orange segments, gorgonzola, and raspberry
vinaigrette.";
```

This code displays the name description of the first special in the two dynamic text fields. It also stops the playhead on the current frame, and registers the device's soft keys.

In the ActionScript layer, select the keyframe in Frame 20 and enter the following code in the Actions panel:

```
stop();
title_txt.text = "Chinese Noodle Salad";
description_txt.text = "Rice noodles with garlic sauce,
shitake mushrooms, scallions, and bok choy.";
```

In the **ActionScript** layer, select the keyframe in **Frame 30** and enter the following code in the **Actions** panel:

```
stop();
title_txt.text = "Seared Salmon";
description_txt.text = "Filet of wild salmon with
    caramelized onions, new potatoes, and caper and tomato
    salsa.";
```

In the **ActionScript** layer, select the keyframe in **Frame 40** and enter the following code in the **Actions** panel:

```
stop();
title_txt.text = "New York Cheesecake";
description_txt.text = "Creamy traditional cheesecake
    served with chocolate sauce and strawberries.";
```

In the **ActionScript** layer, select the keyframe in **Frame 50** and enter the following code in the **Actions** panel:

```
gotoAndStop("specials");
```

This code returns the playhead to the beginning of the animation sequence. The first and last images in the animation sequence are the same, which creates the illusion of a continuous animation.

Save your changes.

Next you'll add navigation to the specials screen that lets the user navigate between images and descriptions of each special.

Open the file you completed in the previous procedure.

In the **Timeline**, select **Frame 10** in the layer named **ActionScript**.

Open the **Actions** panel and enter the following code:

```
Key.removeListener (myListener);
var myListener:Object = new Object ();
myListener.onKeyDown = function () {
    var keyCode = Key.getCode ();
    if (keyCode == ExtendedKey.SOFT1) {
        // Handle left soft key event
        gotoAndPlay ("home");
    }
    else if (keyCode == ExtendedKey.SOFT2) {
        // Handle right soft key event
        play ();
        description_txt.text = "";
        title_txt.text = "";
    }
};
Key.addListener (myListener);
```

The left soft key sends the playhead to the main application screen, and the right soft key advances the image animation to the next image in the sequence.

For more information about using event listeners, see “Using a key listener to handle keypress events” in *Developing Flash Lite 2.x Applications*.

Select Control > Test Movie to test the final application in the emulator.

Create the video screen

In this section, you’ll add video to the application, as well as ActionScript code that lets the user control playback (play, pause, replay) with the right soft key.

In this tutorial, you’ll bundle the device video file in your SWF file’s library, although you can also load an external video file from the device’s file system or a network address.

In Flash, open the file you completed in “[Create the specials screen](#)” on page 27.

Open the Library panel (Window > Library) if not already open.

Open the Library options menu located in the upper-right corner of the Library panel and select New Video.

The Video Properties dialog box appears.

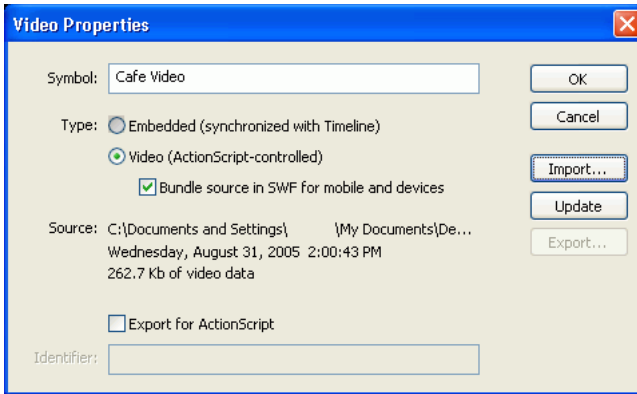
Type a name for the video symbol in the Symbol text box (**cafe Video**, for example).

In the Video Properties dialog box, select the option to bundle the source video in the SWF file, and then click Import.

Open the file named `cafe_townsend_chef.3gp` located at www.adobe.com/go/learn_flt_samples_and_tutorials. On the Samples and Tutorials page, locate, download and decompress the .zip file for your Flash Lite version, and then navigate to the Tutorial Assets folder to access the file.

If you don’t see the video file listed in the Open dialog box (or if you can see it but can’t select it), select All Files (*.*) from the Files of Type pop-up menu (Windows), or All Files from the Enable pop-up menu (Macintosh). This action is sometimes necessary because the Flash authoring tool doesn’t recognize most device video formats.

The Video Properties dialog box should appear as follows before you click OK.

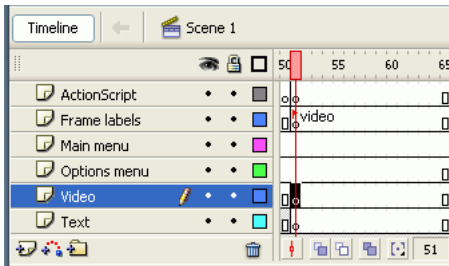


Click OK to close the Video Properties dialog box.

A new video symbol appears in your document's Library panel that is associated with the device video file.

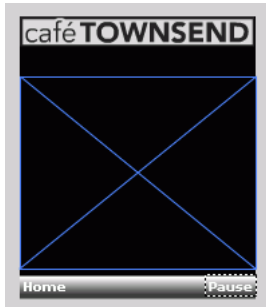
For more information about working with a device video in Flash Lite, see "Using device video" in *Developing Flash Lite 2.x Applications*.

To add the video object to the Stage, in the Timeline select the keyframe in Frame 51 of the layer named Video, as shown in the following example:



From the Library panel, drag the cafe Video object to the Stage.

In the Property inspector, type **caféVideo** in the Instance Name text box, and set the object's *x* position to 0, its *y* position to 45, its Width to 176, and its Height to 144.



In the Timeline, select the keyframe in Frame 51 of the layer named ActionScript.

Open the Actions panel (Window > Actions) and type, or copy and paste, the following code:

```
// Stop timeline, register soft keys, and start video.
stop ();
fscommand2 ("SetSoftKeys", "Home", "Pause");
caféVideo.play ();
var playing:Boolean = true;
// Soft key event handler code:
Key.removeListener (myListener);
var myListener:Object = new Object ();
myListener.onKeyDown = function () {
    var keyCode = Key.getCode ();
    if (keyCode == ExtendedKey.SOFT1) {
        gotoAndPlay ("home");
    }
}
else if (keyCode == ExtendedKey.SOFT2) {
    if (playing) {
        // If video is playing, pause it,
        // set status variable (playing) to false,
        // and set right soft key label to 'Play'.
        caféVideo.pause ();
        playing = false;
        rightSoftKeyLabel.text = "Play";
    }
}
```

```

    }
    else {
        // If video is paused, resume its playback,
        // set status variable (playing) to true,
        // and set right soft key label to 'Pause'.
        caféVideo.resume ();
        playing = true;
        righttSoftKeyLabel.text = "Pause";
    }
}
};
// Register listener object:
Key.addListener (myListener);
//
// Video status handler code.
//
caféVideo.onStatus = function (infoObject:Object) {
    var code = infoObject.code;
    if (code == "completed") {
        // If video has finished playing, set playing=false,
        // and set right soft key label to "Replay":
        playing = false;
        rightSoftKeyLabel.text = "Replay";
    }
};

```

Save your work and test the application in the emulator.

Select the View Video option on the application's home screen to view the video.

Try pausing the video by pressing the right soft key, and pressing the same key again to resume playback. When the video completes, you can press the right soft key again to play the video again.

Create the set location screen

In this section, you'll create a new screen that lets the user select which of the four restaurant locations they want to call for reservations. The location the user selects on this screen determines the number that's dialed when they select Reservations on the application's home screen.

The first time the user starts the application and selects Reservations on the home screen, the application takes them to the set location screen where they can select a location. Subsequently, when the user selects Reservations, the application immediately dials the default restaurant location's number. The application uses a shared object to save the location that the user selected between sessions.

In Flash, open the file you completed in [“Create the video screen” on page 34](#).

In the Timeline, select the keyframe on Frame 66 of the layer named Options Menu.

Open the Library panel (Window > Library), and drag the button named location_SF_button from the library to the Stage.

Position the button under the text that reads, in part, “Select your preferred location...”

In the Property inspector, type **sf_btn** in the Instance Name text box.

Drag the button named location_SJ_button from the library to the Stage and position it directly below the location_SF button.

In the property inspector, type **sj_btn** in the Instance Name text box.

Repeat step 6 for the two buttons in the library named `location_PA` and `location_BK` and give them instance names of `pa_btn` and `bk_btn`, respectively.

The Stage of your application should look something like the following example:



In the Actions panel (Window > Actions), enter the following code:

```
stop ();
fscommand2 ("SetSoftKeys", "Save", "Cancel");
showCurrentLocation();
//
// Soft key event handler code
//
Key.removeListener (myListener);
var myListener:Object = new Object ();
myListener.onKeyDown = function () {
    var keyCode = Key.getCode ();
    if (keyCode == ExtendedKey.SOFT1) {
        // Save location:
        saveNewLocation ();
        gotoAndPlay ("home");
    }
    else if (keyCode == ExtendedKey.SOFT2) {
        // Cancel operation, go back to home screen:
        gotoAndPlay ("home");
    }
};
Key.addListener (myListener);
//
// Function: saveNewLocation().
//
function saveNewLocation () {
    // Determine which button (location) the user selected:
    var selectedButton = Selection.getFocus ();
    switch (selectedButton._name) {
        case "_level0.sf_btn" :
```

```

        // User selected San Francisco.
        location_so.data.phoneNumber = "415-555-1212";
        break;
    case "_level0.sj_btn" :
        // User selected San Jose.
        location_so.data.phoneNumber = "408-555-1212";
        break;
    case "_level0.bk_btn" :
        // User selected Berkeley.
        location_so.data.phoneNumber = "510-555-1212";
        break;
    case "_level0.pa_btn" :
        // User selected Palo Alto.
        location_so.data.phoneNumber = "650-555-1212";
        break;
    }
}
//
// Function: showCurrentLocation().
//
function showCurrentLocation() {
    // Retrieve phone number stored in shared object:
    var phoneNumber:String = location_so.data.phoneNumber;
    // Extract area code from phone number:
    var areaCode:String = phoneNumber.substring(0, 3);
    // Based on area code, set selection focus
    // to corresponding button item:
    switch (areaCode) {
        case "415" :
            Selection.setFocus (_level0.sf_btn);
            break;
        case "408" :
            Selection.setFocus (_level0.sj_btn);
            break;
        case "510" :
            Selection.setFocus (_level0.bk_btn);
            break;
        case "650" :
            Selection.setFocus (_level0.pa_btn);
            break;
    }
}
}

```

Index

A

available Stage size 9

C

Cafe Townsend application
 about 18
 creating navigation 30
 creating the specials animation 26
 creating the specials screen 26
content types 13
creating navigation
 creating a key catcher button 30
 using the soft keys 30

D

dynamic text fields 30

F

Flash Lite
 Adobe Device Central emulator 11
 authoring overview 11
 content types 13

H

Hello World application 9

M

mask layers 26

S

soft keys 30

T

target devices 13
test devices 13
text fields
 dynamic 30
 setting properties 30
tweened animation 26

W

workflow for authoring content 11

