

1

Flash Is Mobile

WHAT'S IN THIS CHAPTER?

- Developing for mobile devices and small screens
- Participants in the mobile ecosystem
- Evolution of the Flash Player optimized for mobile devices
- Core features and architecture of the Flash Lite Player

As a title for the first chapter in the book I thought I'd pay a little homage to one of the early encounters I had with Flash on mobile devices. The title is actually adopted from the Macromedia marketing header during the launch of Flash Lite 2.0 at Macromedia MAX 2005; there have been other noticeable catchy slogans for Adobe's mobile and device marketing since then, such as "Engage with Flash," but here we have a strap line that hits the nail on the head and sets us on our way . . . "*Flash Is Mobile*. . ."

Flash has actually been *mobile* for a number of years now. In order to understand in part how Flash can be utilized on mobile devices, *how* and *where* you, the developer, can leverage the technology, in this chapter you'll take a brief look at the short history and features of Flash Lite Player, the predominant Flash Player runtime used in mobile devices.

MOBILE DEVICES

Since the turn of the century, mobile phones have come a long way from functioning as simple communications devices and today these devices are providing us with everything we need for our daily functions.

Today, mobile devices allow us to listen to our favorite songs, watch recorded video clips and feature films, and record or watch the television shows we have missed. They come with cameras so we can take digital photos and share them with others. We can also use mobiles

to play games and, right now, people all over the world are using their mobile devices to send messages, access the Internet, send and receive email, and use various services daily.

“The mobile phone is the first digital device most people look at in the morning, and the last device before going to sleep at night. It is the only digital device many take to the bathroom, and the only one we carry with us all day.”

—TOMI T. AHONEN AND ALAN MOORE, “Communities Dominate Brands”

The possibilities for the mobile device and its *applications*, right now and beyond, are endless and it is developing mobile applications that is the core focus of this book.

As the underlying technologies used in mobile device handsets have improved, so have the capabilities. In today’s global mobile handset market, consumers have a wide variety of mobile phones to choose from, each with a diverse range of features, aiming to cater for our desires and to make our lives that little bit better.

There are two commonly used terms that categorize the types of mobile phones on the market, the “feature phones” and the “smartphones.” The feature phone is a mobile device with limited or restricted capabilities, where the underlying hardware technology generally tends to be less expensive and at the lower end of the market. Smartphones, on the other hand, are mobile devices with advanced features and capabilities similar to PCs, such as Internet access, and it is this type of mobile device that is becoming more and more popular in the handset market.

For the Flash Lite developer, there is a range of devices that you can target for your applications, and in this book you will focus on developing mobile applications for the smartphone. You’ll cover a wide range of topics relating to Flash Lite mobile application development, through developing a TV Listings Guide, a media Console, an Image Viewer, a Twitter Client, and a Weather Client. All are examples of content that can be consumed by the mobile end user.

Consuming information via a smartphone, anytime, anywhere has never been easier since the introduction of high-speed data networks such as 3G, but some devices are also capable of accessing Wi-Fi networks. But there are several key players involved in ensuring that services and information are delivered to our devices, in a highly competitive, but also thriving, mobile market.

THE MOBILE ECOSYSTEM

Working in the mobile space has its challenges, not just from a developer’s code and development point of view but also for other key participants involved in the mobile ecosystem.

Participants

Figure 1-1 shows the key participants of the mobile ecosystem adapted from Adobe.

The following lists the mobile ecosystem’s key participants, as shown in Figure 1.1; these will be discussed shortly:

- Original Equipment Manufacturers (OEMs)
- Network operators
- Media owners
- Content producers
- Aggregators
- App stores

For each of the key participants mentioned here in the mobile ecosystem, you'll notice that there is a great deal of overlap in goals, and a need for mobile applications to succeed. Flash offers the distinction from other mobile technologies in that it is capable of running on different platforms.

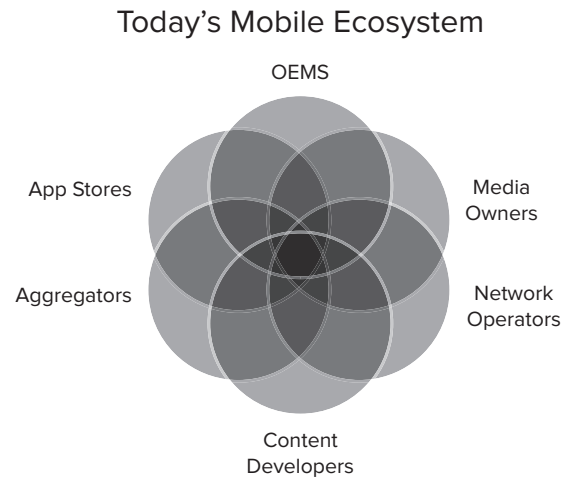


FIGURE 1-1

OEMs

Original equipment manufacturers (OEMs) manufacture the mobile handsets. They are continually developing new products for the market. They are the innovators responsible for fusing the latest hardware with the software platform that each handset model runs on. The OEMs determine the operating system (OS) used by the device, which can also determine the device's features and capabilities. OEMs work directly with network operators to ensure that networking hardware is also integrated. For OEMs, it's imperative to have software on the device that can engage the end user and, essentially, distinguish their products from others in the market.

Network Operators

Network operators manage the network adopted by the consumer. The operators provide the services and channels of content that are accessed by their subscribers. Operators are also known as *carriers*. They are always looking for ways to improve services to bring more subscribers to their networks.

Media Owners

Media owners effectively own the rights to distribute particular content. Media owners want to be able to distribute their brands to as many domains as possible. Like the OEMs, media owners want to distinguish their brands and ensure that they stand out from the crowd.

Content Producers

Content producers represent the designers and developers who produce mobile content. They often work with the brand guidelines from the media owners to design, develop, and produce mobile content.

Aggregators

Aggregators syndicate and distribute mobile content through online portals and also to the network operators ready for consumption by the consumer. Aggregators were early adopters of syndicating

mobile content, who saw great potential of collecting mobile content developed by third-party content producers and distributing it.

App Stores

Unless you've been living in a cave for the past year, you know that app stores are where consumers can find content to install on their mobile phones. With devices such as the iPhone, apps, applications, and app stores have been a huge commercial success.

As a Flash mobile developer, the opportunities to generate engaging experiences on mobile phones are vast. Innovative Internet services and applications that require data services are at the heart of the mobile ecosystem and are two of the driving forces behind mobile usage.

Recommended Reading

For a detailed breakdown of the mobile ecosystem, I recommend that you read the Strategy Analytics PDF “*Understanding the Mobile Ecosystem*” .pdf file found on the Adobe Web site at http://www.adobe.com/devnet/devices/articles/mobile_ecosystem.pdf.

ENGAGING EXPERIENCES THAT WORK ON SMALL SCREENS

Flash has changed the way that people consume and engage with content on the Internet around the world. From animation to video to rich media platforms, today Flash continues to push the boundaries for our digital world and is establishing itself as the platform for the next generation of publishing across multiple screens.

Developing for mobile devices using Flash requires a slightly different train of thought when compared to desktop or Internet browsers:

- **Screen size** — Developing for smaller screens will mean you have to think more *inside of the box* as well as outside it; in particular, you will need to pay more attention to the detail provided in your applications.
- **Processing power** — With most smartphones having a lower or limited CPU power compared to home PCs, performance measurements need to be taken into consideration.
- **Interaction and navigation** — There is no mouse on a mobile device. The majority of smartphones have keypads; some have touch screens; others have both, which means that, during development, you have to really carefully consider *how* the end user interacts with the Flash application on the device.
- **Memory** — Ultimately mobile devices tend to have less available memory than desktop computers. Regardless of what Flash applications you develop, they will have an impact on memory, and you will need to be aware of the memory consumed by your applications and the limits to which you can go.

If you are already a mobile developer, then no doubt these issues are already familiar to you, and you know that developing for mobile is not the same as developing for the desktop or Web browsers on personal computers.

What is Flash Lite?

Adobe describes Flash Lite as an “optimized version” of the Flash Player software that is used in PC’s for desktop and Web browsers. The optimized runtime particularly addresses aspects of content running on devices with limited capabilities.

The .swf (SWF) file extension is synonymous with Flash and all versions of the Flash Player, including Flash Lite. The SWF format was introduced in 1998, and the Flash Player is the runtime engine that executes bytecode contained within the .swf files. You’ll be making many a SWF throughout the course of this book.

At the time of writing, there are over 700 types of devices supporting the Flash Player. Flash Lite has been installed on more than one billion devices and can be directly distributed to millions of open OS smartphones.

So what’s the story on Flash Lite?

Past, Present, and Future

In what now seems like *“a long time ago in a galaxy far, far away . . .”*, the software company Macromedia (acquired by Adobe) saw the potential of content-rich media on mobile devices. With its early release of Flash Lite in Korea, it achieved its goal of replicating Flash content, produced on personal computers, on other devices, albeit with limited features compared to those on the browser- and desktop-based Flash Player.

The browser-based versions of the Flash Player have always been out of sync with the mobile version, because of the limitations of the device hardware.

Flash Lite 1.x

Flash had already been introduced to Pocket PC devices by the time Flash Lite first appeared as Flash Lite 1.0 on the Japanese NTT DoCoMo i-mode system, which offered mobile Web sites and content such as animations, games, wallpapers, and screensavers for consumers to download over their network. Flash Lite 1.0 was then superseded by Flash Lite 1.1, a release that had a number of enhancements, including network access and integration with the handset.

Flash Lite 1.1, allowed existing Flash developers, who created similar content for Web sites, to leverage their skills. After a few years of progress with Flash on mobile in the Asia Pacific (APAC) region, the Flash Lite 1.1 player made its way to developers in other regions of the world: Europe, the Middle East, and Africa (EMEA); North America, and South America. This iteration of the Flash Lite Player signaled the start of a new dawn for Flash on mobile devices, and as the developer community became more established, licensing agreements were made with OEMs to pre-install the Flash Lite Player on mobile handsets allowing consumers to view compatible content.

Flash Lite 2.x

A few years after the release of 1.1, Flash Lite 2.0 was introduced with performance improvements and new features, including the ability to play both device video and sound, attracting further

interest from OEMs and establishing the player as a fully rich media player for the mobile market among its competitors.

Developers from other mobile software development backgrounds, attracted by the ease of using and rapid development possible with 2.0, also found Flash Lite to be a great interactive mobile solution.

The open source application development platform established by Qualcomm BREW (Binary Runtime Environment for Wireless) later was given a slightly different implementation of Flash Lite, in Flash Lite 2.1.

Flash Lite 3.x

Not long ago Flash 3.0 opened up new avenues for Flash on mobile. Fueled in part by a “revolution” in Flash video on the Internet, the more evolved features in version 3.0 meant that the player would allow playback of video directly into the Flash Player, another boost for the multimedia capabilities of the player.

Other features of the player included performance enhancements and the ability to stream media through Adobe Flash Media Server (FMS). More recently Flash Lite 3.1.x became the first iteration of the Flash Lite Player that can be distributed with content created by developers.

This book primarily covers Flash Lite 3.x. There is another version of the Flash Lite Player currently in development that will be backwardly compatible; however, at the time of writing, this is not available to developers.

The Open Screen Project

The Open Screen Project (<http://www.openscreenproject.org/>) is an industry-wide collaboration to establish a “consistent runtime environment” for experiences across a variety of screens and devices. The initiative recognizes the Adobe Flash platform as the key technology to deliver consistency and provide consumers with the rich experiences they demand, whether in a browser or on the desktop, a mobile device, TV, or a gaming console. This is an ambitious project, one that remains central to Adobe, which leads the initiative, and one that will also be central to the strategies of the majority of content owners in the not too distant future.

Flash Player 10.1

At the time of writing, the next increment of the Adobe Flash Player runtime, Flash Player 10.1, is going through its public beta for developers. This is the first runtime release adopted by the Open Screen Project, which enables Flash content to run across numerous platforms and devices, including mobile. This is the first full Flash player targeted at both mobile devices and PCs; the announcement of the release signals that mobile devices are becoming more powerful, capable of presenting content that matches that of high-end desktop technology.

Flash Lite 3.x content will be supported in both Flash Player 10.1 and the next “unreleased” version of the Flash Lite player, and it is said that devices containing this are soon to be referred to simply as “Flash enabled.”

Multiple Devices and Platforms

Although the sole focus of this book is on developing mobile applications, you should be aware the Flash Lite Player 3.x has been ported to a variety of consumer electronics and numerous platforms, including:

- Nokia
- Sony Ericsson
- HTC
- Microsoft
- Chumby
- iRiver

At the same time that the adoption of Flash Player 10.1 will undoubtedly grow in 2010, the integration of the Flash Lite Player into devices will also continue to grow for a while and is estimated to peak in the billions by the end of 2010.

OEMs Supporting Flash Lite Player Development

Device manufacturers Nokia and Sony Ericsson are among a number of OEMs supporting Flash mobile development, and these two in particular have resources that you can utilize in your Flash development. One of the very early debates among developers during the early release of Flash Lite was whether Flash Lite or Java Platform, Micro Edition (JME) would prevail as the mobile technology of choice. They coexist, and Sony Ericsson has enabled developers to leverage the benefits of both technologies in the Project Capuchin initiative, which has enabled the Flash Player to be bundled in a J2ME wrapper to provide the additional features of a device; this also provides a good crossover between the two software entities. Extending Flash Lite applications with Project Capuchin is covered in Chapter 12.

Types of Flash Lite Mobile Content

One of the challenging aspects of mobile application development is the complexity of devices, operating systems, and capabilities of each device. The key advantage of Flash is that it can run on multiple handsets and operating systems. However, not all content types will support all Flash Lite features; even if the Flash Lite version is the same, there still could be some aspect of Flash Lite that isn't supported.

Flash Lite content comes under particular categories, including:

- Screensavers
- Ringtones
- Standalone applications

This book contains content specifically created for the *standalone* versions of Flash Lite.

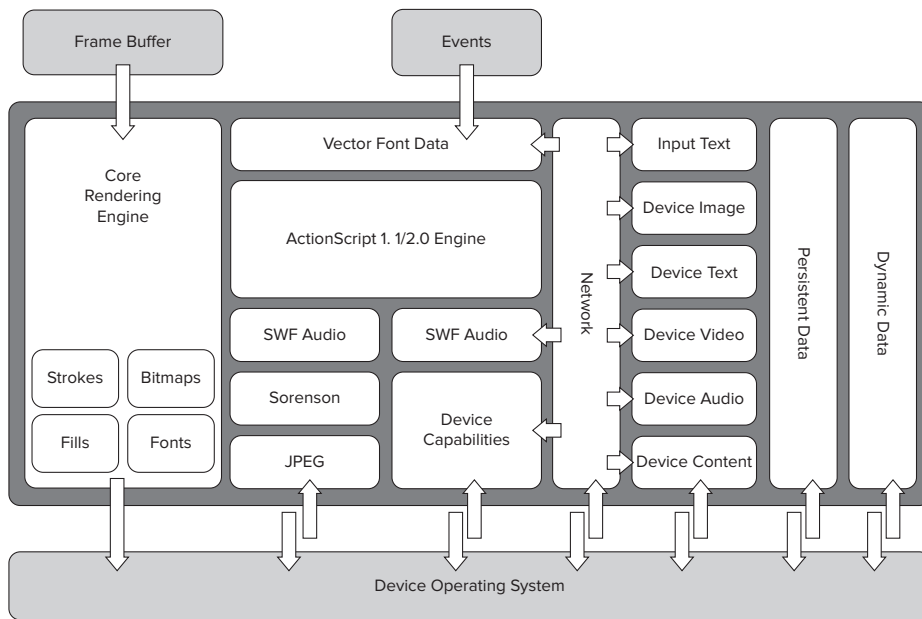
Adobe Device Central provides a more comprehensive list of content types that you can create using Flash Lite. It also provides the full set of properties and features supported by various mobile device handsets. You'll learn more about Device Central in Chapter 2.

The aim of each application covered in the book is to provide you with practical solutions for developing a wide range of Flash applications for mobiles devices.

FLASH LITE ARCHITECTURE

In this section, we'll take a look at the Flash Lite 3.x Player architecture and go over the features that you'll be covering in the next section.

In Figure 1-2, you can see the overall architecture of the Flash Lite 3.0 player.



SOURCE: [HTTP://WWW.ADOBE.COM/PRODUCTS/FLASHLITE/ARCHITECTURE/](http://www.adobe.com/products/flashlite/architecture/)

FIGURE 1-2

The main aspects of the Flash Lite runtime can be seen as:

- Core Rendering Engine
- ActionScript 1.1/2.0 Engine
- Vector Font Data
- SWF Audio, a component enabling audio to play back
- Sorenson

- JPEG
- Device Capabilities
- Device Content, Audio, Video, Text, and Image
- Persistent Data
- Dynamic Data

You'll cover aspects of the architecture in the next section.

Flash Lite Player 3.x Features

The majority of the Flash Lite Player features shown in the architecture diagram are covered throughout the chapters of this book:

- ActionScript 2.0
- Device features and capabilities
- Loading external resources
- Video support
- Audio support
- Font and text support
- Images
- Persistent data
- Dynamic data and XML support

Flash Lite gives the Flash mobile platform a lot of features, such as ActionScript 2.0 for accessing and controlling device keys and functions, networks and connectivity, audio playback, event and streaming sound, different types of texts, fonts, navigation controls, and images.

ActionScript 2.0

Although ActionScript 3.0 (AS3) has been around for a while now, Flash Lite 3.x fully supports the ActionScript 2.0 (AS2) engine. Based on the ECMA 262 standard, AS2 is a modern programming language that supports dot-syntax and object-oriented development. This book fully covers examples written in AS2.

You'll cover AS2 development in Chapter 3.



At the time of writing this book, the beta for the Flash Player 10.1 using version AS3 was announced. There is currently no developer edition of Flash Lite that supports AS3.

Device Features and Capabilities

Every mobile device handset has its own particular features and capabilities, implemented by the OEMs. Video playback, sound support, and predictive user input text are all examples of features that may or may not be supported across a different range of mobile devices.

With the Flash Lite 3.x Player, you have ways of determining specifically which features or capabilities are supported by the device dynamically through AS2 properties and functions:

- `System.capabilities`
- `fscommand()` and `fscommand2()`

Both these functions allow you to target aspects of your applications during development. By using `System.capabilities`, you can access the properties of the device, such as determining the language. By using *fscommands* you can access and adjust device features directly from your mobile applications, such as monitoring the battery and network signal levels, sending text messages, making phone calls, altering the device's backlight, and setting the device to vibrate.

You'll learn more about accessing device features and capabilities in Chapter 3.

OEMs Nokia and Sony Ericsson also have APIs specifically for developing Flash Lite applications for their supported handsets, which allow you to access more advanced features of their devices and extend your Flash Lite applications. Features such as sending files via Bluetooth, gaining access to the Accelerometer, GPS or the device's file systems are examples of features which need to be retrieved by the Flash Lite Player.

You'll learn more about extending Flash Lite applications in Chapters 10 and 11.

Loading External Resources

Network connectivity enables Flash Lite 3.x applications to dynamically load data and resources, that reside externally, into the main SWF application. Multimedia, text-based content, and data objects are all resources supported externally from folders residing on the actual device or from a Web server over HTTP.

You'll learn more about loading external resources across the chapters in this book.

Audio Support

Flash Lite 3.x supports a range of file formats for audio playback, including MIDI, MP3, Pulse Code Modulation (PCM), Adaptive Differential PCM (ADPCM), and SMAF.

You can embed sounds directly into the Flash Lite application or load them into the application over HTTP or from a location on the device. Streaming audio via RTMP using Adobe Flash Media Server (FMS) is also supported.

You'll learn all about using sound in your applications in Chapter 7.

Video Support

Flash Video (FLV) is widely regarded as the “*most popular video format*” on the Internet. Support for FLV via the On2 VP6 and Sorenson video codecs, was one of the key features introduced in

Flash Lite 3.0 allowing for the format to be played directly within the Flash Lite Player. The Flash Lite Player for mobile now includes support for H.264, the standard used in high definition (HD) televisions, game consoles, and video players.

Flash Lite also supports video playback of other video formats supported by the majority of mobile devices, including 3GPP and MPEG-4 files.

Video files can be embedded directly into the Flash Lite application, and they can also be loaded into the application externally over a HTTP server or from a location on the device itself at runtime. Delivery of video can also be made over the Real Time Messaging Protocol (RTMP) using media servers such as FMS and Red5.

You'll learn all about how to include video in your applications in Chapter 7.

Text and Font Support

Developing applications for Flash Lite allows you to include a variety of text components in your applications, including dynamic text fields, text for user input, and static text. Static fields are created during authoring time, whereas dynamic text and input text can be created during authoring time and at runtime.

You also have a choice between using device fonts or embedded fonts. With each choice there is a tradeoff; embedded fonts give you more design control of content but reduce the amount of memory available to the application. Using device fonts gives you less control over design but provide the reassurance that the content will render text.

Flash Lite 3.x supports the UTF-8 character set, including complex languages such as Chinese, Thai, Arabic, and Hebrew. For devices that support the features, Flash Lite also allows predictive text and inline text input. Text measurement and text wrapping are also supported by text fields and device-specific vector fonts for improved small text readability.

Navigation and Keys

The Flash Lite Player also accesses user input events from the device, including the keys for navigation up, down, left, right, and select, and also text, character symbols, and number keys from 0 to 9, *, and #, for example. Figure 1-3 shows an example of device keys supported on the Nokia E71 (from Device Central).

You'll cover navigation and interaction in Chapter 3.

Images

Using Flash Lite, you can render image files directly in your applications including PNG, JPEG, and GIF files. You'll cover including images in your applications in Chapter 8.



FIGURE 1-3

Persistent Data

With Flash Lite you can also save data to the device and retrieve it for later use in your applications. You'll cover saving data in your applications in Chapter 8.

Dynamic Data and XML Support

Also available in Flash Lite 3.x is the ability to load and parse dynamic data, including native classes for loading and parsing of external XML. Several chapters throughout the book cover utilizing dynamic data and parsing XML in particular.

SUMMARY

In this chapter we wanted to give you an overview of the Flash Lite technology and provide a little background as to why the technology is used and where it is headed.

You took a look at the key participants in the mobile ecosystem and also gained insight into various aspects of the Flash Lite architecture, covering the core features of the Flash Lite 3.x Player. You've now learned what types of content are supported on devices with the Flash Lite Player installed, and at this stage you should have an idea of what you can develop with Flash Lite. In the upcoming chapters you will learn how to implement many of the features that the player supports, which were highlighted in this chapter.

The next chapter will take you through *getting started* with Flash Lite development.