

## 1

# WHAT IS WEBKIT?

**WEBKIT IS AN** open-source rendering engine designed to display web pages. It powers Google Chrome and Safari as well as a variety of mobile devices such as iPhone, iPad, and Android phones and tablets. It also controls rendering in a variety of desktop applications such as Apple Mail and those created with Adobe AIR.

WebKit basically consists of two components: WebCore and JavaScriptCore. WebCore controls all layout and rendering tasks, creating HTML visualizations based on DOM markup and

Cascading Style Sheets (CSS). JavaScriptCore compliments this layout engine with a high-performance interaction engine that provides feature-rich JavaScript functionality.

This chapter provides a short history of WebKit and a discussion of which browsers, devices, and applications use the engine. You learn the various components that WebKit supports, as well as how these differ across various implementations.

### QUICK HISTORY

The code that would later become WebKit began in 1998 as part of KDE's web browser, Konqueror (designed primarily for Linux).

As part of an effort to build their own web browser, Apple forked the HTML and JavaScript engine from KDE in 2002, naming the ported engine *WebKit*. Initially Apple and KDE cross-developed the engines, exchanging code patches and enhancements. Then Apple made a number of unilateral decisions, including converting a portion of the codebase from C++ to Objective C, and dramatically altering other coding approaches. This along with Apple's source control and poor documentation led to a split between the KDE project and WebKit in 2005, which some KDE developers characterized as a "bitter failure."

In June 2005, Apple announced that it was open-sourcing the entirety of WebKit (previously only WebCore and JavaScriptCore were open-source). This was reportedly part of an effort to work "more closely with KDE and the rest of the open-source community," according to Safari developer Maciej Stachowiak.

As part of the Safari web browser, WebKit gained notoriety in the web community by being the first browser to support a number of CSS3 attributes. While other browsers offered more limited rendering tools, Safari provided a variety of innovative styling controls such as rounded corners and gradients that could generate richer graphics in CSS.

Even as other browsers began supporting these attributes, WebKit-based browsers stayed ahead of the curve by implementing new features, such as CSS animations and 2D and 3D transformations. These impressive features provided a robust rendering toolkit, allowing developers to smoothly animate, rotate, and skew HTML elements using CSS alone. These WebKit innovations proved so useful that they eventually informed the W3C CSS3 recommendations, and became standards used across all modern browsers today.

### WHERE IS WEBKIT USED?

WebKit's speed, power, and versatility make it an ideal rendering solution for many situations. The engine is used in a wide variety of implementations, from desktop browsers to mobile devices and web-enabled applications.

### WEB BROWSERS

An incredibly feature-rich and fast rendering engine, WebKit is ideal for desktop web browsers. It has been adopted by Google Chrome and Safari as well as a variety of niche browsers.

At its inception, WebKit represented a very small portion of the browser market share because it was powering only the relatively underused Safari and Konqueror. Although Safari usage is increasing, it still comprises a modest 7% of the market, and Konqueror's share is still negligible.

WebKit's share of the browser market climbed considerably when it was adopted by the Google Chrome project. Between Chrome and Safari, WebKit comprises 28% of the desktop browser market (at the time of this writing).

## MOBILE DEVICES

Recently Mobile WebKit, a special build of WebKit, has become the emerging standard for mobile device browsers. Lightweight yet surprisingly powerful, Mobile WebKit is an ideal engine for creating rich applications and web experiences on underpowered mobile devices.

The vast majority of smartphones built in recent years use the WebKit framework. Apple iOS devices such as iPhone, iPad and iPod are all built upon WebKit. Additionally, WebKit is used by most Android devices, which include a wide variety of phones and tablets. Newer Blackberry devices, such as the Blackberry Torch, as well as Symbian-based devices use WebKit. Also, WebOS, which powers the Palm Pre and a number of tablets, is also based on WebKit, however at the time of this writing WebOS' future is uncertain since it has been abandoned by HP.

## RIAS

In addition to web browsers and mobile devices, WebKit is also used in a number of desktop applications. Due to its roots as a web browser, the vast majority of these apps are rich Internet applications (RIAs).

WebKit powers apps built with Adobe AIR, a cross-platform framework that allows developers to simultaneously program applications for Windows, Mac, and Linux.

WebKit also is used in Mac OS X dashboard widgets, as well as a number of miscellaneous Mac and Windows apps, such as Apple's Mail and Microsoft's Entourage.

## MISCELLANEOUS DEVICES

The same features that make WebKit ideal for mobile devices also make it an attractive option for other underpowered devices such as eReaders and video game consoles.

A variety of reading tablets such as the Kindle and Nook use WebKit to offer a stripped-down web browser on their devices. The Nintendo 3DS, a portable video game console, offers a WebKit-based browser as well.

As more devices become web enabled, you will likely see them adopting WebKit as a powerful open-source solution to integrate a web layer.

## WHAT DOES WEBKIT SUPPORT?

The WebKit engine is known for supporting a wide variety of advanced functionality. WebKit leads the pack when it comes to browser support for innovative new features in CSS3, HTML5, and JavaScript.

### CSS3

WebKit is perhaps best known as the first major engine to support a variety of CSS3 attributes. These features provide developers with an advanced toolkit for rendering complex web pages without relying on images. WebKit supports rounded corners, gradients, drop shadows, custom fonts, and alpha transparency, to name a few. Additionally, it supports CSS animations both through transitions as well as keyframes. Furthermore, WebKit is constantly innovating new experimental attributes and pushing CSS standards into the future.

WebKit also supports a variety of advanced CSS3 selectors, allowing you to choose elements based on their HTML attributes and location in the DOM. You can select direct descendants and siblings, or even the fifth paragraph in a `div`. You can leverage the `:before` and `:after` pseudo-elements to inject content directly into the page, and apply special styling based on the state of elements.

You can even take advantage of CSS media queries to tailor styling to different screen dimensions. That means you can create flexible layouts, which change in response to smaller and larger screen sizes.

### HTML5 AND SVG

WebKit also supports a number of advanced features in HTML5. For instance, media tags allow you to embed audio and video directly in the browser, as you learn in Chapter 6. Furthermore, a variety of new form elements can be used to provide a richer user interface without relying on JavaScript. Additionally, these elements can be coupled with HTML5 form validation to ensure the user fills required fields, enters the correct information, and sees appropriate error messages.

WebKit supports canvas as well as SVG, enabling you to draw complex vectored shapes directly in the browser. Combining these drawings with JavaScript interaction, you can build rich applications that rival those built in Flash. With the introduction of WebGL, you can use these elements to create stunning 3D applications.

### NOT ALL WEBKITS ARE CREATED EQUAL

One common misconception about WebKit is that the engine is identical across all implementations. Applications not only use different versions of WebKit, but there are additionally a number of modified versions of the core.

### USING DIFFERENT VERSIONS OF WEBKIT

You probably expect certain changes across different versions of a single browser, or between browsers that use completely different renderers. But even the latest versions of different WebKit browsers do not necessarily use the same version of the engine.

Although browsers typically use a recent build of WebKit, different browser release schedules lead to disparities at any given time. For instance Apple took more than a year between releasing Safari versions 5 and 5.1. Throughout this time, the WebKit engine in Safari slowly became outdated compared to that of Chrome, which had more frequent releases.

Furthermore, browsers often make decisions about which parts of WebKit to support. For example, the default configuration of Chrome enables WebGL, which is also enabled in recent versions of Safari if the user has Mac OS X version 10.6 or higher. Thus the user's operating system can affect how much WebKit functionality is supported in his browser.

## MODIFYING THE CORE

As mentioned earlier, the WebKit engine is open source, which has led to a variety of modified versions. For instance, Mobile WebKit differs substantially from the standard core, which means that web apps can behave differently between Desktop and iOS Safari.

Mobile implementations of WebKit even differ from device to device. For instance, Nokia ported the WebKit core to create its Symbian operating system, and Palm used WebKit to create an interface for WebOS, both of which differ from Mobile WebKit.

This means that you will notice incongruities between different implementations of WebKit, which can prove much more complicated than disparities between different versions of a single browser.

## SUMMARY

WebKit began with open-source, standards-driven principles at its core, and these values continue to shape its development today. From improving its already lightweight and powerful core, to innovating new standards in web development, you can count on WebKit remaining a dominant force amongst web browsers.

Constant refinement of the library ensures that future platforms will continue to view WebKit as the ideal option for web rendering. Furthermore, creating new standards in CSS, JavaScript and HTML shapes the future of not only WebKit-based browsers, but all other browsers on the market.

