

Schnelleinstieg in Adobe AIR

von Hans Weigl

Adobe AIR ist eine betriebssystemübergreifende Laufzeitumgebung, die eine völlig neue Art von Rich-Internet-Applikationen möglich macht: AIR-Anwendungen können sowohl auf Daten aus dem Web als auch auf den Desktop des Users zugreifen – und das auf der Basis gängiger Webtechnologien wie HTML, CSS, JavaScript, Flash und Flex.

In diesem TecFeed lernen Sie anhand von konkreten Beispielen die Struktur und Funktionsweise von AIR-Anwendungen kennen und erfahren, wie Sie sie debuggen, kompilieren und eine AIR-Installationsdatei packen. Der Zugriff auf das lokale Verzeichnissystem, die Einbindung von PDF-Dateien und SQLite werden ebenso behandelt.

Schnelleinstieg in Adobe AIR richtet sich an Leser, die Kenntnisse in einer Programmiersprache mitbringen und sich mit HTML, JavaScript und ActionScript auskennen.

O'REILLY®

INHALT

Einführung | 2

AIR im Überblick | 3

Durchstarten mit AIR | 6

AIR und HTML | 12

Die File-API von AIR | 27

Fenstertechniken | 39

Paralipomena | 50

Was die Zukunft bringt | 56

Über den Autor | 59



TecFeeds

www.tecfeeds.de

Einführung

Was ist AIR?

AIR ist eine betriebssystemübergreifende Laufzeitumgebung, die es Entwicklern ermöglicht, mit HTML und CSS, JavaScript und Ajax, Flash und Flex Rich-Internet-Applikationen (RIAs) zu erstellen, die auch auf dem Desktop einsetzbar sind. AIR-Anwendungen laufen im Flash Player und können Daten verwenden, die sowohl im Web als auch auf dem eigenen Desktop zu finden sind. Auf diese Weise wird es möglich, ohne Browser auf Online-Quellen und lokale Dateien zuzugreifen und so den Übergang vom Desktop ins Web für den User verschwinden zu lassen.

Ursprünglich als Alphaversion unter dem Codenamen *Apollo* veröffentlicht, liegt inzwischen die Beta der Software vor, die nun unter dem Namen *Adobe Integrated Runtime* (AIR) läuft. Mit dieser Version wollen wir uns hier intensiv beschäftigen. AIR ist frei verfügbar und kann in den Adobe Labs (<http://labs.adobe.com>) heruntergeladen werden.

An wen richtet sich dieser TecFeed?

Dieser *Schnelleinstieg in Adobe AIR* richtet sich an Leser, die Grundkenntnisse in der Programmierung mitbringen und ohne viel Umschweife in die Anwendungsentwicklung mit AIR einsteigen wollen. Darüber hinaus sollten Sie Kenntnisse in HTML, JavaScript und etwas mit ActionScript 3 mitbringen und die Installation der AIR-Laufzeitumgebung sowie des einen oder anderen Editors nicht scheuen. Die Beispiele sind

nicht klassenbasiert formuliert oder gar in mehrere Quell-dateien aufgeteilt. Es ist immer eine einzige kleine (MXML- oder HTML-)Datei, die den Text weniger ergänzt, als dessen Dreh- und Angelpunkt ist.

Was Sie erwartet

Kapitel 1, *AIR im Überblick*, gibt einen Überblick über Adobe AIR und skizziert die Besonderheiten von AIR. Kapitel 2, *Durchstarten mit AIR*, stellt die verschiedenen IDEs vor, die Sie mit AIR nutzen können, und behandelt die Dateistruktur von AIR-Anwendungen, ihre Distribution und ihr Update-Management. Kapitel 3, 4 und 5 bilden das Herzstück dieses TecFeeds. In Kapitel 3, *AIR und HTML*, starten wir mit einem HTML-basierten AIR-Beispiel, das Sie im einfachsten Texteditor selbst editieren und verändern können. Wir erklären Ihnen, welche Dateien für eine AIR-Anwendung unbedingt erstellt werden müssen und wie diese aufgebaut sind. Anschließend zeigen wir Ihnen, wie Sie das Ganze debuggen, kompilieren und zu einer AIR-Installationsdatei packen können. In Kapitel 4, *Die File-API von AIR*, führen wir Sie in die API ein, die es ermöglicht, dass Ihre AIR-Anwendung in das Dateien- und Verzeichnis-system des Rechners eingreifen kann. Gerade diese Fähigkeit reiht eine AIR-Anwendung nahtlos in Ihre üblichen Desktop-Anwendungen mit ein. In Kapitel 5, *Fenstertechniken*, widmen wir uns den verschiedenen Fenstertechniken, mit denen Sie Ihrer AIR-Anwendung ein individuelles Gesicht verleihen können. Natürlich wollen wir Themen wie PDF oder SQLite nicht unter den Tisch fallen lassen. Dazu haben wir für Sie in

Kapitel 6, *Paralipomena*, einiges vorbereitet. Kapitel 7 schließlich gibt einen Ausblick darauf, was zukünftig von Adobe AIR zu erwarten ist.

Der Beispielcode

Der Beispielcode zu diesem TecFeed liegt auf der Verlagswebseite unter www.oreilly.de/catalog/pdf_adobeairger zum Download für Sie bereit. In dem Archiv finden Sie neben dem Code, der in diesem PDF behandelt wird, einen Hinweis, mit welchem Editor wir das Beispiel bearbeitet haben.

Mein Dank geht an ...

... die Fachgutachter, allen voran Sascha Kersken, einen bekannten Fachbuchautor, dessen umfassendes Publikationsspektrum von Flash über Ruby bis hin zu Linux und Apache reicht. Besonderer Dank geht an Inken Kiupel von O'Reilly, die das Projekt ermöglicht und mit Rat und Tat von Anfang bis Ende begleitet hat. Schließlich und vor allem geht mein tiefer Dank an Sandra, die ich ewig vernachlässigend als meinen steten Quell der Inspiration weiß. Zuletzt danke ich allen Bits und Bytes, aus denen das kleine Werk besteht.

Hans Weigl, München 2007

AIR im Überblick

Web ohne Browser: die neue Interneterfahrung

Wenn Sie heute ans Internet denken, stellen Sie sich sicherlich ein geöffnetes Browserfenster vor, das mehr oder weniger gelungene Webseiten zeigt. Wie die Webpages angezeigt werden, hängt unglücklicherweise auch noch von den Fähigkeiten des benutzten Browsers ab. Das wird in Zukunft anders. Auch wenn die Tage des klassischen Browsers beileibe nicht gezählt sind, entsteht mit der Laufzeitumgebung AIR doch eine Alternative, die multilingual mit JavaScript, HTML und Flash (allmählich auch mit PDF) umgehen kann und besonders für RIAs (Rich Internet Applications) die richtige Wahl ist.

Adobe hat mit AIR ein Projekt ins Leben gerufen, das in zweierlei Hinsicht bemerkenswert ist: Zum einen ist es damit möglich, Webentwicklungen auf dem Desktop laufen zu lassen, und zum anderen lassen sich mit AIR viele Features problemlos umsetzen, die typisch sind für Web 2.0-Anwendungen. Erinnern Sie sich an die sieben Punkte aus der Definition von Tim O'Reilly (nachzulesen unter <http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html>): Einer davon lautet *Rich User Experiences*. Das erreichen Sie fast mühelos durch Flash-basierte AIR-Anwendungen. Ein anderer lautet: *Software Above the Level of a Single Device*. Diesen Punkt interpretieren wir hier so, dass Sie mit AIR in einem Arbeitsgang nicht nur Mac- und Windows-Software schreiben, sondern diese mit ein paar Hand-

griffen auch wieder in eine Browseranwendung zurückverwandeln können. Man müsste hinsichtlich AIR einen weiteren Punkt ergänzen, für den Guy Kawasaki auf der Gnomedex 2007 den passenden Begriff geprägt hat: AIR-Entwicklung sei »Web-Enabled Desktop Development«, im Ergebnis erhalte man also *webfähige betriebssystemunabhängige Desktop-Software*.

Ein prominentes AIR-Projekt

Ein besonders viel diskutiertes AIR-Projekt ist *San-Dimas* (<http://www.projectsandimas.com/>) von eBay, das den Handel mit eBay in die Form einer AIR-Anwendung bringt. Auch wenn die Features von *San-Dimas* anfangs sicher bescheiden sein werden, vermuten wir hier eine kreative Alternative zu der eher hölzernen Webpage, die die Kundenbindung verstärkt.

AIR ist eine betriebssystemübergreifende Laufzeitumgebung, die frei in den Adobe Labs unter <http://labs.adobe.com> verfügbar ist. Natürlich liegt die AIR-Laufzeitumgebung nicht so tief im Betriebssystem, dass man darauf bevorzugt ein Desktop-basiertes Office-Paket entwickeln würde. Man sollte sich bewusst sein, dass AIR im Wesentlichen dazu dient, RIAs für den Desktop zu bauen. Technisch realisiert man AIR-Anwendungen am einfachsten in der Flex 3-Entwicklungsumgebung, einer IDE, in der nicht nur die großartigen Möglichkeiten von Flash stecken.

Unterstützte Betriebssysteme

Unterstützung finden zurzeit die Betriebssysteme MS-Windows XP mit SP2, MS-Vista Home Premium Edition (und höher) und Macintosh Mac OS X 10.4 auf PPC oder Intel.

AIRs Datei-I/O

AIRs Filesystem-API verleiht den Anwendungen die Fähigkeit, lokal mit Dateien bzw. Verzeichnissen zu arbeiten, wie man es vom Desktop her gewohnt ist. So lassen sich Dateien sowie Verzeichnisse erzeugen, kopieren, verschieben oder löschen. Man kann ebenso ganze Verzeichnisse auslesen, auflisten und alle möglichen Informationen über die enthaltenden Dateien vom Betriebssystem beziehen. Die Serialisierung/Deserialisierung von ActionScript-Objekten ist damit möglich sowie das Auslesen und Schreiben von Binärdateien (wie z.B. Audio- oder Bilddateien).

AIR spricht viele Sprachen

AIR integriert zwei primäre Webtechniken: Flash und HTML. Mit den APIs von Flash Player 9 steht das ECMAScript-basierte ActionScript 3 zur Verfügung. Darüber hinaus wurden einige dieser APIs für AIR entsprechend erweitert. Da die Flex 2-IDE auf ActionScript 3 basiert, lassen sich aus Flex 2 heraus sehr bequem AIR-Anwendungen entwickeln. Die HTML-Unterstützung von AIR beinhaltet im Detail neben HTML auch XHTML, JavaScript, CSS und das Document Object Model (DOM). Das bedeutet, dass es ebenso möglich ist, AIR-Anwendungen ganz ohne Flash zu bauen.

AIR UND FLASH

Der Flash Player verleiht AIR die Fähigkeit, ActionScript in Echtzeit zu interpretieren, mit multimedialen Inhalten wie Bildern, Sound und Video umzugehen und über ein vollständig vektorbasiertes Rendering zu verfügen. Genauso werden die Protokolle HTTP und RTMP unterstützt, und der Aufbau von XML-Sockets und Binär-Sockets ist ebenfalls möglich.

AIR UND HTML

Die in AIR verwendete HTML-Engine ist die in KHTML und Safari verwendete Open Source-Engine Webkit (<http://www.webkit.org>). Webkit und Flash Player sind auf einem so tiefen Level miteinander verknüpft, dass HTML wie ein Flash-Display-Objekt behandelt werden kann (also manipulierbar ist wie z.B. eine Bitmap).

AIR-Anwendungen können tatsächlich komplett aus HTML und JavaScript bestehen, wenn hier die Kompatibilität zu Webkit gewährleistet ist. Dass AIRs HTML-Rendering auf Webkit aufsetzt, bedeutet auch, dass JavaScript, das in AIR läuft, ebenso im Safari- und im Firefox-Browser laufen wird. Bezüglich des Internet Explorer (IE) muss man diese Aussage einschränken, da beispielsweise XMLHttpRequests im IE anders formuliert werden müssen.

AIR ALS BRÜCKE ZWISCHEN ACTIONSCRIPT UND JAVASCRIPT

Die angesprochene tiefe Verbindung von Webkit und Flash Player birgt darüber hinaus das Potenzial, zwischen ActionScript (in SWF-Dateien) und JavaScript (in HTML) eine wechselseitige

seitige Kommunikation aufbauen zu können. Zum Beispiel kann JavaScript ActionScript-Events wahrnehmen und umgekehrt. Uns stehen mehrere Möglichkeiten zur Verfügung:

- ActionScript in einer SWF-Datei kann das HTML-DOM von geladenem HTML-Code (als Child-Objekt im SWF) lesen und verändern.
- Analog kann ActionScript aus einer geladenen HTML-Seite JavaScript-Variablen lesen und setzen sowie JavaScript-Methoden aufrufen.
- Geladenes JavaScript hat Zugriff auf alle ActionScript-Klassenobjekte und -methoden im es beinhaltenden (parent-)SWF. Es kann sogar neue Klassenobjekte instantiiieren (wie z.B. ein `new Sound()`- oder `new Socket()`-Objekt).

Im Abschnitt »Die Kommunikation zwischen JavaScript und ActionScript« in Kapitel 3 zeigen wir dazu ein Beispiel, das diese wechselseitige Kommunikation illustriert.

Dokumenttechniken in AIR

So vielfältig wie die programmiertechnischen Fähigkeiten von AIR sind auch die Möglichkeiten bezüglich der Darstellung und Manipulation elektronischer Dokumente. Neben HTML als Dokumenttechnologie soll es in späteren Versionen von AIR volle Unterstützung für das PDF-Format geben.

Auf den Geschmack gekommen

Von dieser allgemeinen Charakterisierung angespornt, wollen wir uns ohne Umschweife in medias res begeben und im fol-

genden Kapitel die AIR-Laufzeitumgebung installieren sowie die Struktur von AIR-Anwendungen studieren. Bis dahin!

Durchstarten mit AIR

Installation der Laufzeitumgebung

Damit Sie ohne Zeitverlust mit der Entwicklung von AIR-Anwendungen beginnen können, müssen Sie vorher die AIR-Laufzeitumgebung auf Ihrem Rechner installieren, die dafür sorgt, dass Ihre AIR-Applikationen dort laufen können. Dazu gehen Sie schrittweise vor :

Laden Sie sich den Installer http://download.macromedia.com/pub/labs/air/air_b1_win_061107.exe (Windows) bzw. http://download.macromedia.com/pub/labs/air/air_b1_mac_061107.dmg für Mac OS von den Adobe Labs herunter und starten Sie anschließend den Installationsvorgang.

Falls die vom Flash Player benutzte Datei *mms.cfg* (die *Macromedia Security Configuration*-Datei) modifiziert wurde, sollten Sie sie entfernen, während AIR-Anwendungen laufen. Einige der Einstellungen in dieser Datei könnten die AIR-Anwendungen in ihrer Funktionalität einschränken. Sie finden diese Datei bei einem Windows-Rechner standardmäßig im Windows-Verzeichnis unter `\system32\macromed\flash\mms.cfg` oder beim Mac unter `/Library/Application Support/Macromedia/mms.cfg`.

Beim Mac wird eine Anmeldung als Admin verlangt, damit die Installation ausgeführt wird. Geben Sie in diesem Fall Ihren Mac-Benutzernamen und das Benutzerpasswort ein.



Es ist ratsam, eine früher installierte Apollo-Version (die frühere Alphaversion von AIR) zu deinstallieren und erst anschließend die neue Betaversion von AIR zu installieren. Vorhandene Apollo-Anwendungen werden, nachdem die neue AIR RTE (Run Time Environment) installiert ist, nicht mehr laufen.

Ist die Hürde der Installation genommen, wollen Sie vielleicht spontan eine AIR-Applikation ausprobieren. Adobe bietet unter <http://labs.adobe.com/wiki/index.php/Apollo:Applications:Samples> eine Reihe von AIR-Samples an, die Sie sofort installieren können, wenn Sie die angebotenen gepackten Installationsfiles (mit der Endung *.air*) herunterladen und deren Installation anschließend per Doppelklick starten.

Eine andere interessante AIR-Anwendung ist *Finetune* (<http://www.finetune.com/desktop/>), ein Musikdienst der besonderen Art. Zum Starten der Installation klicken Sie auf der Website einfach auf *Install Now*. Nun wird das *air*-Paket geladen und meldet sich im Anschluss mit einem Prompt. Klicken Sie auf *Open*, im nächsten Fenster auf *Install* und anschließend auf *Continue*. Sehen Sie nun den Button *Finish*, dann ist alles erfolgreich verlaufen. Klicken Sie zum Start von *Finetune* auf diesen Button, und der Player öffnet sich (siehe Abbildung 1).

Unter Windows können Sie nun über das Startmenü unter *Programme/Finetune* die Applikation (die jetzt in der Tat eine ausführbare Binärdatei ist) starten. Alternativ dazu wurde während der Installation auch ein Programmsymbol auf dem Desktop angelegt. Auf einem Mac können Sie *Finetune* mit