

IDŹ DO

PRZYKŁADOWY ROZDZIAŁ



SPIS TREŚCI

KATALOG KSIĄŻEK

KATALOG ONLINE

ZAMÓW DRUKOWANY KATALOG

TWÓJ KOSZYK

DODAJ DO KOSZYKA

CENNIK I INFORMACJE

ZAMÓW INFORMACJE
O NOWOŚCIACH

ZAMÓW CENNIK

CZYTELNIA

FRAGMENTY KSIĄŻEK ONLINE

Adobe AIR dla programistów JavaScript. Leksykon kieszonkowy

Autor: Mike Chambers, Daniel Dura,
Kevin Hoyt, Dragos Georgita

Tłumaczenie: Grzegorz Werner

ISBN: 978-83-246-1960-3

Tytuł oryginału: [Adobe AIR for JavaScript
Developers Pocket Guide](#)

Format: 115x170, stron: 200



Poznaj moc nowoczesnej technologii AIR w tworzeniu funkcjonalnych aplikacji internetowych!

- Jak skonfigurować środowisko programistyczne w AIR?
- Jak zaimplementować funkcje?
- Jak utworzyć aplikację AIR, wykorzystując HTML i JavaScript?

Adobe AIR to nowe międzyplatformowe środowisko wykonawcze, które pozwala programistom na wykorzystanie znanych technologii, takich jak HTML i JavaScript, do tworzenia rozbudowanych aplikacji internetowych. AIR oferuje bogaty zbiór wywołań API, a także ścisłą integrację z pulpitem, co umożliwia łączenie potęgi zasobów lokalnych z dostępem do internetu. O atrakcyjności Adobe AIR świadczą więc zarówno zasięg i łatwość programowania (najlepsze cechy modelu WWW), jak i funkcjonalność oraz bogactwo modelu pulpitu.

Książka „Adobe AIR dla programistów JavaScript. Leksykon kieszonkowy” stanowi wprowadzenie do tworzenia aplikacji Adobe AIR z wykorzystaniem języków HTML i JavaScript. Leksykon zawiera przegląd mechanizmów wykonawczych HTML i JavaScript w Adobe AIR oraz wyjaśnia, jak uzyskać dostęp do funkcji i interfejsów Adobe AIR z poziomu JavaScript. Znajdziesz tu wszelkie porady i wskazówki niezbędne do tego, by skonfigurować środowisko programistyczne, stworzyć, przetestować i wdrożyć atrakcyjną oraz funkcjonalną aplikację WWW.



Spis treści

Przedmowa	5
1. Wprowadzenie do Adobe AIR	18
Krótka historia aplikacji WWW	18
Problemy z dostarczaniem aplikacji za pośrednictwem przeglądarki	21
Przegląd Adobe AIR	23
Podstawowe technologie Adobe AIR	24
2. Pierwsze kroki w Adobe AIR	37
Co jest potrzebne do tworzenia aplikacji Adobe AIR?	37
Odinstalowywanie wstępnych wersji Adobe AIR	39
Instalowanie Adobe AIR	41
Odinstalowywanie Adobe AIR	42
Konfigurowanie Adobe AIR SDK i narzędzi wiersza poleceń	42
Tworzenie prostej aplikacji AIR przy użyciu HTML-a i JavaScriptu	48
Testowanie aplikacji	55
Pakowanie i wdrażanie aplikacji AIR	59
3. Praca z językami JavaScript i HTML w Adobe AIR	67
WebKit w Adobe AIR	67
JavaScript w Adobe AIR	70
Implementacja funkcji w AIR	71

Model bezpieczeństwa	79
Używanie szkieletów JavaScriptu	87
Rozwiązywanie problemów z aplikacjami AIR	98
4. Minireceptury Adobe AIR	103
Wdrażanie aplikacji	103
Styl aplikacji	108
Obsługa okien	111
Plikowy interfejs API	121
Wybieranie plików	140
Monitorowanie usługi i serwera	147
Online-offline	152
Przeciąganie i upuszczanie	155
Osadzona baza danych	161
Argumenty wiersza polecenia	174
Sieć	176
Dźwięk	185
A Aliasy JavaScriptu	187
Skorowidz	193

Rozdział 2. Pierwsze kroki w Adobe AIR

W tym rozdziale wyjaśnimy, jak zacząć tworzenie aplikacji Adobe AIR z wykorzystaniem HTML-a i JavaScriptu. Omówimy:

- instalację Adobe AIR;
- konfigurowanie Adobe AIR SDK i narzędzia wiersza poleceń;
- tworzenie pierwszej aplikacji AIR;
- testowanie aplikacji AIR;
- podpisywanie, pakowanie i wdrażanie aplikacji AIR.

Pod koniec tego rozdziału czytelnicy będą dysponowali prawidłowo skonfigurowanym środowiskiem do tworzenia aplikacji AIR. Będą też wiedzieli, jak buduje się, testuje i wdraża aplikacje Adobe AIR.

Co jest potrzebne do tworzenia aplikacji Adobe AIR?

Aby przystąpić do tworzenia aplikacji AIR, trzeba dysponować kilkoma elementami.

Środowisko uruchomieniowe Adobe AIR

Środowisko uruchomieniowe Adobe AIR jest potrzebne do testowania ikon oraz wdrażania aplikacji Adobe AIR. Środowisko można pobrać bezpłatnie pod adresem: <http://www.adobe.com/go/getair>.

Adobe AIR SDK

Pakiet Adobe AIR SDK zawiera narzędzia wiersza poleceń, przykładowe pliki i inne zasoby, które ułatwiają tworzenie aplikacji AIR.

W szczególności będziemy używać narzędzi wiersza poleceń dołączonych do SDK (ADL i ADT), które umożliwią testowanie, podpisywanie i pakowanie aplikacji AIR z poziomu praktycznie dowolnego środowiska programistycznego.

AIR SDK można pobrać bezpłatnie pod adresem: <http://www.adobe.com/go/getairsdk>.

Zintegrowane środowisko programistyczne lub edytor HTML/JavaScript

Budowanie aplikacji AIR przy użyciu HTML-a i JavaScriptu wymaga jakiegoś sposobu tworzenia plików HTML i JavaScript. Można użyć dowolnego narzędzia, które pozwala na tworzenie i edycję plików tekstowych (takiego jak VIM lub Notatnik), choć warto posłużyć się narzędziem ułatwiającym pracę z plikami HTML i JavaScript, takim jak Adobe Dreamweaver, Coda albo Aptana Studio.

UWAGA

Więcej informacji o programie Adobe Dreamweaver można znaleźć pod adresem: <http://www.adobe.com/go/dreamweaver>; o Codzie firmy Panic — pod adresem: <http://www.panic.com/coda/>; a o Aptana Studio — pod adresem: <http://www.aptana.com>.

Obsługiwane systemy operacyjne

Choć aplikacje AIR można tworzyć i pakować w praktycznie dowolnym systemie operacyjnym (także w Linuksie), ich testowanie i wdrażanie jest możliwe tylko w systemach, które są obsługiwane przez Adobe AIR.

Obsługiwane systemy operacyjne to:

- Microsoft Windows 2000 SP4,
- Microsoft Windows XP SP2,
- Windows Vista Home Premium, Business, Ultimate lub Enterprise,
- Mac OS 10.4.910 i nowsze wersje (procesory Intel i PowerPC).

UWAGA

Odtwarzanie wideo H.264 w systemie Mac wymaga procesora Intela.

Adobe obecnie pracuje nad wersją Adobe AIR dla Linuksa.

Odinstalowywanie wstępnych wersji Adobe AIR

Ci, którzy zainstalowali wstępne wersje Adobe AIR (alfa lub beta), być może będą musieli odinstalować je przed zainstalowaniem wersji 1.0. Nie zawsze jest to jednak konieczne.

W tabeli 2.1 wymieniono wstępne wersje Adobe AIR i wyjaśniono, czy trzeba je odinstalować przed zainstalowaniem wersji 1.0.

Tabela 2.1. Wstępne wersje Adobe AIR

Wersja	Odinstalować?
Alfa	Odinstalować
Beta 1	Zostanie odinstalowana automatycznie przez instalator Adobe AIR 1.0
Beta 2	Nie trzeba odinstalowywać
Beta 3	Nie trzeba odinstalowywać

Jak widać w tabeli 2.1, wersje beta 2 i beta 3 mogą działać obok Adobe AIR 1.0. Umożliwia to uruchamianie aplikacji zbudowanych pod kątem wersji beta 2 i beta 3, dopóki nie skończy się ważność tych wersji.

Ci, którzy nie wiedzą, czy zainstalowali wersję alfa, na wszelki wypadek powinni odinstalować wszystkie wstępne wersje Adobe AIR.

Odinstalowywanie w Windows

Aby odinstalować wszystkie wstępne wersje Adobe AIR w systemie Windows, należy wykonać poniższe czynności:

1. Z menu *Start* wybrać polecenie *Ustawienia/Panel sterowania*.
2. Kliknąć dwukrotnie ikonę *Dodaj lub usuń programy*.
3. Wybrać wstępną wersję Adobe AIR, która ma zostać odinstalowana (w zależności od wersji może ona nosić pseudonim „Apollo”).
4. Kliknąć przycisk *Zmień/usuń*.

Odinstalowywanie w systemie Mac

Aby odinstalować wszystkie wstępne wersje Adobe AIR w systemie Mac, należy wykonać opisane niżej czynności. W zależności od zainstalowanych wersji niektóre etapy będzie można pominąć.

1. Uruchomić program Adobe AIR Uninstaller w katalogu */Users/<użytkownik>/Applications* (gdzie *<użytkownik>* jest nazwą konta w systemie).
2. Uruchomić program Adobe AIR Uninstaller w katalogu */Applications*.
3. Usunąć katalog */Library/Frameworks/Adobe Apollo.framework*.

4. Usunąć plik */Library/Receipts/Adobe Apollo.pkg*.

5. Opróżnić kosz.

Po wykonaniu tych czynności można przystąpić do instalacji wersji 1.0 środowiska uruchomieniowego.

Instalowanie Adobe AIR

Choć tworzenie i testowanie aplikacji Adobe AIR nie wymaga instalowania środowiska Adobe AIR, warto to zrobić, aby wypróbować inne aplikacje AIR oraz przetestować pakowanie i wdrażanie własnej aplikacji.

Instalowanie środowiska uruchomieniowego jest proste. Wystarczy pobrać i uruchomić instalator Adobe AIR. W tym celu należy:

1. Pobrać instalator AIR pod adresem: <http://www.adobe.com/go/getair>.
2. Uruchomić instalator. W Macu trzeba najpierw zamontować plik *.dmg*, który zawiera program instalacyjny.
3. Postępować według wskazówek instalatora.

UWAGA

Można również zainstalować Adobe AIR bezpośrednio z poziomu środowiska uruchomieniowego przy użyciu instalacji ekspresowej. Opiszemy to w rozdziale 4.

Adobe AIR to środowisko uruchomieniowe, a nie aplikacja, którą można wykonać. Żeby sprawdzić, czy zostało zainstalowane prawidłowo, najprościej będzie zainstalować aplikację AIR. W tym celu można pobrać i zainstalować dowolną aplikację AIR albo samodzielnie zbudować prostą aplikację AIR według wskazówek zamieszczonych w dalszej części rozdziału.

UWAGA

Przykładowe aplikacje AIR można pobrać z witryny firmy Adobe pod adresem: http://www.adobe.com/go/air_samples.

Odinstalowywanie Adobe AIR

Proces odinstalowywania Adobe AIR zależy od używanego systemu operacyjnego.

Odinstalowywanie w Windows

W Windows można odinstalować Adobe AIR w taki sam sposób jak każdą inną aplikację. Wystarczy wybrać pozycję *Adobe AIR* w oknie *Dodawanie lub usuwanie programów* w Panelu sterowania.

Odinstalowywanie w Macu

Instalator Adobe AIR umieszcza program deinstalacyjny w systemie użytkownika. Aby odinstalować Adobe AIR, należy uruchomić program o nazwie *Adobe AIR Uninstaller*, który znajduje się w katalogu */Applications/Utilities*.

Konfigurowanie Adobe AIR SDK i narzędzi wiersza poleceń

Adobe AIR SDK zawiera narzędzia i próbki kodu, które ułatwiają tworzenie, testowanie i wdrażanie aplikacji. W szczególności zawiera dwa narzędzia wiersza poleceń, którymi będziemy się posługiwać:

ADL

To narzędzie służy do uruchamiania i testowania aplikacji Adobe AIR bez jej uprzedniego instalowania.

ADT

To narzędzie służy do pakowania i podpisywania aplikacji AIR w celu przygotowania jej do dystrybucji.

Instalowanie Adobe AIR SDK

Aby ułatwić sobie pracę, należy umieścić te pliki na ścieżce systemowej. Dzięki temu będzie można uruchamiać narzędzia z dowolnego katalogu.

Narzędzia wiersza poleceń znajdują się w katalogu *bin* pakietu SDK. W celu zainstalowania pakietu należy wykonać poniższe czynności:

1. Pobrać Adobe AIR SDK pod adresem: <http://www.adobe.com/go/getairsdk>.
2. Otworzyć pakiet SDK:
 - a. W Windows — zdekompresować archiwum ZIP.
 - b. W Macu — zamontować plik *.dmg*.
3. Skopiować zawartość SDK do systemu (będziemy określać tę lokalizację mianem *<ścieżka_SDK>*).

UWAGA

Aby uruchamiać narzędzia wiersza poleceń, wystarczy skopiować katalogi *bin*, *lib* i *runtime* z pakietu SDK.

Należy zachować względne położenie ścieżek do katalogów *bin*, *lib* i *runtime*.

4. W tym momencie na dysku powinny znajdować się przynajmniej trzy następujące katalogi: *<ścieżka_SDK>/bin*, *<ścieżka_SDK>/lib* oraz *<ścieżka_SDK>/runtime*. Narzędzia ADL i ADT znajdują się w katalogu *bin*.

Umieszczanie narzędzi wiersza poleceń na ścieżce systemowej

Teraz wystarczy umieścić katalog `<ścieżka_SDK>/bin` na ścieżce systemowej, aby móc uruchamiać aplikacje wiersza poleceń z dowolnego miejsca w systemie.

Sposób wykonania tego zadania zależy od używanego systemu operacyjnego (Windows lub Mac).

Windows

Użytkownicy Windows powinni wykonać następujące czynności:

1. Otworzyć okno dialogowe *Właściwości systemu* (można je znaleźć w Panelu sterowania) i kliknąć zakładkę *Zaawansowane*.
2. Kliknąć przycisk *Zmienne środowiskowe*.
3. Zaznaczyć pozycję *PATH* i kliknąć przycisk *Edytuj*. Dodać ścieżkę do katalogu *bin* na końcu bieżącej wartości zmiennej, oddzielając ją średnikiem:

```
; <ścieżka_sdk>/bin
```

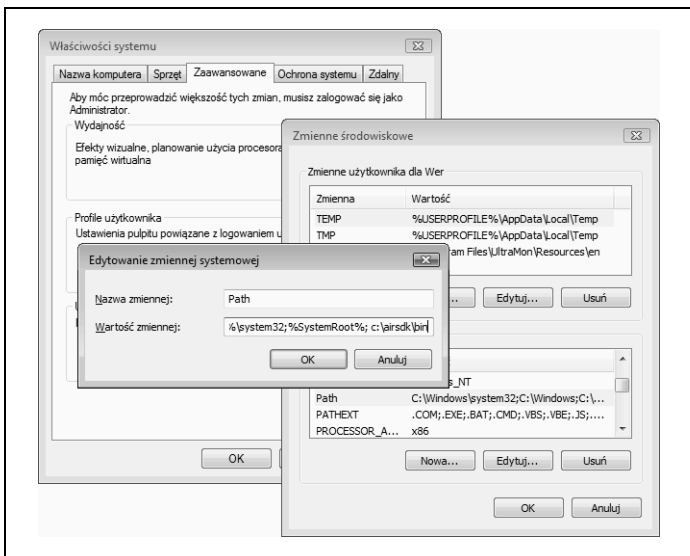
Edytowanie ścieżki w Windows pokazano na rysunku 2.1.

4. Kliknąć przycisk *OK*, aby zamknąć okno dialogowe.

Aby przetestować instalację, należy otworzyć nową konsolę Windows (*Start/Uruchom/cmd*) i wpisać polecenie **adt**.

UWAGA

Należy otworzyć nowe okno konsoli, aby zostały uwzględnione nowe ustawienia zmiennej *PATH*.



Rysunek 2.1. Umieszczanie narzędzi wiersza poleceń na systemowej ścieżce w Windows

Powinny pojawić się następujące wyniki:

```
usage:
  adt -package SIGNING_OPTIONS <air-file> <app-desc>
FILE_ARGS
  adt -prepare <air-file> <app-desc> FILE_ARGS
  adt -sign SIGNING_OPTIONS <air-file> <air-file>
  adt -checkstore SIGNING_OPTIONS
  adt -certificate -cn <name> ( -ou <org-unit> )?
( -o <org-name> )? ( -c <country> )?
<key-type> <pfx-file> <password>
  adt -help
```

```
SIGNING_OPTIONS: -storetype <type> ( -keystore <store> )?
( -storepass <pass> )? ( -keypass <pass> )? ( -providerName
<name> )? ( -tsa <url> )?
FILE_ARGS: <fileOrDir>* ( ( -C <dir> <fileOrDir>+ ) |
( -e <file> <path> ) ) * -C dir
```

Oznacza to, że narzędzia są skonfigurowane prawidłowo.

Jeśli pojawi się komunikat, że nie znaleziono pliku, należy:

- Upewnić się, że katalogi *bin*, *lib* i *runtime* znajdują się w katalogu `<ścieżka_SDK>`.
- Upewnić się, że ścieżka do katalogu `<ścieżka_SDK>` jest dołączona do zmiennej środowiskowej `PATH`.
- Upewnić się, że przed wydaniem polecenia zostało otwarte nowe okno konsoli.

Mac

Istnieje kilka sposobów dodawania AIR SDK do ścieżki systemowej, w zależności od używanej powłoki oraz metody określania zmiennych użytkownika.

Poniżej wyjaśniono, jak zmodyfikować zmienną środowiskową `PATH` w powłoce `bash`. W tym celu należy wykonać poniższe czynności:

1. Otworzyć program Terminal (*/Applications/Utilities/Terminal*). Przejść do katalogu macierzystego, wpisać polecenie `cd` i nacisnąć klawisz *Enter*.
2. Wydać polecenie `ls -la`, aby sprawdzić, czy obecny jest jeden z dwóch plików z ustawieniami powłoki.
3. Poszukać pliku o nazwie `.profile` lub `.bashrc`.
4. Jeśli nie ma żadnego z tych plików, utworzyć plik `.profile` za pomocą polecenia `touch .profile`.
5. Otworzyć plik `.profile` lub `.bashrc` w edytorze tekstu.
6. Poszukać wiersza o mniej więcej takiej zawartości:

```
export PATH=$PATH:/usr/local/bin
```

7. Na końcu tego wiersza dodać ścieżkę do katalogu `<ścieżka_SDK>/bin`. Jeśli na przykład ścieżką do katalogu `bin` jest `/airsdk/bin`, polecenie eksportu powinno wyglądać mniej więcej tak:

```
export PATH=$PATH:/usr/local/bin:/airsdk/bin
```

Poszczególne pozycje ścieżki powinny być rozdzielone dwukropkami.

8. Jeśli plik jest pusty, dodać do niego następujący wiersz:

```
export PATH=$PATH:/airsdk/bin
```

9. Zapisać i zamknąć plik.

10. Wydać polecenie **source .profile** (albo **source .bashrc**, jeśli zmodyfikowano plik `.bashrc`), aby wczytać nowe ustawienia.

11. Sprawdzić, czy nowe ustawienia zostały uwzględnione, przez wpisanie **echo \$PATH** i naciśnięcie klawisza *Enter*. Ścieżka do katalogu `<ścieżka_SDK>/bin` powinna pojawić się w wynikach polecenia.

12. W celu przetestowania instalacji otworzyć okno terminala i wydać polecenie **adt**.

Powinny pojawić się następujące wyniki:

```
usage:
adt -package SIGNING_OPTIONS <air-file> <app-desc>
FILE_ARGS
adt -prepare <airi-file> <app-desc> FILE_ARGS
adt -sign SIGNING_OPTIONS <airi-file> <air-file>
adt -checkstore SIGNING_OPTIONS
adt -certificate -cn <name> ( -ou <org-unit> )?
( -o <org-name> )? ( -c <country> )?
<key-type> <pfx-file> <password>
adt -help
```

```
SIGNING_OPTIONS: -storetype <type> ( -keystore
<store> )? ( -storepass <pass> )? ( -keypass <pass> )?
```

```
( -providerName <name> )? ( -tsa <url> )? FILE_ARGS:  
<fileOrDir>* (( -C <dir> <fileOrDir>+ ) | ( -e <file>  
<path> ))* -C
```

Oznacza to, że narzędzia są skonfigurowane prawidłowo.

Jeśli pojawi się komunikat, że nie znaleziono pliku, należy:

- Upewnić się, że katalogi *bin*, *lib* i *runtime* znajdują się w katalogu `<ścieżka_SDK>`.
- Upewnić się, że ścieżka do katalogu `<ścieżka_SDK>` jest prawidłowo dołączona do zmiennej środowiskowej `PATH`.
- Upewnić się, że przed wydaniem polecenia zostało otwarte nowe okno terminala albo że przetworzono plik konfiguracyjny poleceniem `source`.

Tworzenie prostej aplikacji AIR przy użyciu HTML-a i JavaScriptu

Po zainstalowaniu Adobe AIR i Adobe AIR SDK jesteśmy gotowi do napisania pierwszej aplikacji AIR.

Utworzymy bardzo prosty przykład: „Witaj, świecie”. Pomyślnie zbudowanie i przetestowanie aplikacji będzie oznaczać, że nasze środowisko programistyczne jest prawidłowo skonfigurowane do budowania bardziej złożonych i funkcjonalnych aplikacji AIR.

Tworzenie plików aplikacji

Każda aplikacja Adobe AIR składa się przynajmniej z dwóch plików. Pierwszy to główny plik treści. Jest to podstawowy plik HTML lub SWF, który zostanie wyświetlony (wykonany) zaraz po uruchomieniu aplikacji.

Drugi plik jest nazywany plikiem deskryptora aplikacji. Jest to plik XML, który zawiera metadane aplikacji używane przez środowisko Adobe AIR.

Utwórzmy te pliki na użytek naszej aplikacji. W tym celu należy:

1. Utworzyć nowy folder o nazwie *AIRHelloWorld*.
2. W tym folderze utworzyć dwa nowe pliki o nazwie *AIRHelloWorld.html* i *AIRHelloWorld.xml*.
3. Otworzyć oba pliki w ulubionym edytorze tekstu lub kodu HTML (albo w środowisku IDE).

Pliki deskryptora aplikacji

Plik deskryptora aplikacji to plik XML, który jest wymagany przez każdą aplikację Adobe AIR. Zawiera on ogólne metadane (takie jak nazwa i opis aplikacji) używane przez Adobe AIR, a także informacje dotyczące sposobu, w jaki aplikacja powinna być uruchamiana. Określa się w nim główny plik aplikacji oraz początkowy tryb wyświetlania okna.

Najpierw przyjrzymy się całemu plikowi deskryptora (*AIRHelloWorld.xml*) naszej aplikacji, a następnie omówimy jego poszczególne elementy.

UWAGA

Przykładowy plik deskryptora aplikacji znajduje się w pakiecie AIR SDK w folderze *templates*.

Należy otworzyć plik *AIRHelloWorld.xml* i wpisać następujący tekst:

```
<?xml version="1.0" encoding="utf-8" ?>
<application
xmlns="http://ns.adobe.com/air/application/1.0">

  <id>com.oreilly.AIRHelloWorld</id>
  <filename>AIRHelloWorld</filename>
```



```

<name>AIR Hello World</name>
<description>Prosta aplikacja AIR</
description> <version>1.0</version>

<initialWindow>
  <content>AIRHelloWorld.html</content>
  <title>AIR Hello World</title>
  <systemChrome>standard</systemChrome>
  <transparent>false</transparent>
  <visible>true</visible>
  <minimizable>true</minimizable>
  <maximizable>true</maximizable>
  <resizable>true</resizable>
</initialWindow>
</application>

```

Treść pliku jest dość oczywista, ale omówimy ją wiersz po wierszu, aby dokładnie zrozumieć znaczenie poszczególnych elementów.

```
<application xmlns="http://ns.adobe.com/air/application/1.0">
```

Przeźren nazw określa wersję Adobe AIR przyjętą przez aplikację — w tym przypadku 1.0.

Przyjrzyjmy się następnemu elementowi:

```
<id>com.oreilly.AIRHelloWorld</id>
```

Element `id` jest ważny; określa on unikatowy identyfikator aplikacji. Adobe AIR używa identyfikatorów do odróżniania poszczególnych aplikacji.

Jak widać, użyto odwrotnego formatu domenowego, który czytelnicy mogą znać z języków programowania takich jak Java, ActionScript i niektóre szkielety JavaScript. Można utworzyć własny identyfikator z nazwy aplikacji i nazwy domeny.

Następna sekcja elementów określa ogólne metadane aplikacji:

```

<filename>AIRHelloWorld</filename>
<name>AIR Hello World</name>
<description>Prosta aplikacja AIR</
description> <version>1.0</version>

```

Poszczególne elementy są opisane w tabeli 2.2.

Tabela 2.2. Elementy metadanych aplikacji

Element	Opis
filename	Nazwa natywnego pliku aplikacji, który zostanie utworzony.
name	Nazwa aplikacji. Ta nazwa będzie widoczna dla systemu operacyjnego i dla użytkownika.
description	Element opcjonalny. Opis aplikacji, który zostanie pokazany użytkownikowi podczas procesu instalacji.
version	Element wymagany. Określa wersję aplikacji (taką jak „1.0”, „v1” itd.).

Następny element to tag `initialWindow`, który zawiera elementy określające sposób wykonania aplikacji przez środowisko uruchomieniowe:

```
<initialWindow>
  <content>AIRHelloWorld.html</content>
  <title>AIR Hello World</title>
  <systemChrome>standard</systemChrome>
  <transparent>>false</transparent>
  <visible>>true</visible>
  <minimizable>>true</minimizable>
  <maximizable>>true</maximizable>
  <resizable>>true</resizable>
</initialWindow>
```

Element `content` jest wymagany i wskazuje główny plik aplikacji, w tym przypadku plik HTML.

UWAGA

Plik deskryptora aplikacji i główny plik treści muszą znajdować się w tym samym folderze.

Element `initialWindow` zawiera kilka innych elementów, które określają początkowe parametry okna oraz styl aplikacji w momencie jej uruchomienia (tabela 2.3).

Tabela 2.3. Elementy `initialWindow`

Element	Opis
<code>title</code>	Tytuł głównego okna aplikacji. Element opcjonalny.
<code>systemChrome</code>	Styl systemowy, którego powinna używać aplikacja (<code>standard</code> lub <code>none</code>).
<code>transparent</code>	Wskazuje, czy tło aplikacji powinno być przezroczyste. Ta opcja ma zastosowanie tylko wtedy, gdy element <code>systemChrome</code> ma wartość <code>none</code> . Element opcjonalny. Wartość domyślna to <code>false</code> .
<code>visible</code>	Wskazuje, czy aplikacja będzie widoczna zaraz po uruchomieniu. Przydaje się, jeśli aplikacja musi przeprowadzić jakąś skomplikowaną inicjalizację, zanim wyświetli interfejs użytkownika. Element opcjonalny. Wartość domyślna to <code>false</code> .
<code>minimizable</code>	Wskazuje, czy aplikacja może być minimalizowana. Element opcjonalny. Wartość domyślna to <code>true</code> .
<code>maximizable</code>	Wskazuje, czy aplikacja może być maksymalizowana. Element opcjonalny. Wartość domyślna to <code>true</code> .
<code>resizable</code>	Wskazuje, czy można zmieniać rozmiar okna aplikacji. Element opcjonalny. Wartość domyślna to <code>true</code> .

W naszym przykładzie użyjemy stylu okien systemu operacyjnego.

UWAGA

Zauważmy, że element `visible` ma domyślną wartość `false`. Oznacza to, że jeśli bezpośrednio nie ustawimy go na `true`, uruchomiona aplikacja nie będzie miała interfejsu użytkownika. W takim przypadku trzeba będzie ustawić właściwość `visible` na `true` metodą programową.

Może to być przydatne, jeśli aplikacja zaraz po uruchomieniu musi przeprowadzić jakąś skomplikowaną inicjalizację lub rozmieścić elementy. W takim przypadku można pozwolić jej na wykonanie tych operacji i dopiero wtedy wyświetlić interfejs użytkownika.

To wszystko, co jest potrzebne w pliku deskryptora naszej aplikacji. Teraz jesteśmy gotowi do utworzenia głównego pliku HTML aplikacji.

Tworzenie głównego pliku aplikacji

Główny plik aplikacji jest wczytywany w momencie uruchamiania aplikacji. Może to być albo skompilowany plik Flash (SWF), albo plik HTML.

W tym rozdziale utworzymy bardzo prosty plik HTML, aby sprawdzić, czy nasze środowisko programistyczne jest skonfigurowane prawidłowo. Bardziej zaawansowane zastosowania AIR API zostaną omówione w rozdziałach 3. i 4.

```
<html>
<head>
  <title>AIRHelloWorld</title>

  <script>
    function init()
    {
      runtime.trace("wywołano funkcję init");
    }
  </script>
</head>
<body onload="init()">
  <div align="center">Witaj, świecie</div>
</body>
</html>
```

Jak widać, jest to bardzo prosty plik HTML, który po wczytaniu i inicjalizacji wyświetla napis „Witaj, świecie” oraz wywołuje funkcję JavaScriptu.

Warto zwrócić uwagę na kilka wierszy:

```
<body onload="init()">
```

Ten wiersz wskazuje, że używamy standardowego zdarzenia `onload` elementu `body` jako punktu wejścia do kodu JavaScript naszej aplikacji.

```
<script>
    function init()
    {
        ...
    }
</script>
```

W tych wierszach używamy standardowej funkcji JavaScriptu do przechwycenia zdarzenia `onload`.

Dostęp do wywołań API Adobe AIR

W funkcji JavaScriptu znajduje się kod, który może nie być znajomy:

```
runtime.trace("wywołano funkcję init");
```

Jest to jedyny specyficzny dla AIR kod w całej aplikacji. Właściwość `runtime` jest dołączana do obiektu okna przez Adobe AIR i zapewnia punkt wejścia do mechanizmu Adobe AIR oraz jego wywołań API. Funkcja `trace` to wywołanie API najwyższego poziomu, które przyjmuje łańcuch i wyświetla go w wierszu poleceń (jeśli aplikacja została uruchomiona z wiersza poleceń).

W języku JavaScript dostęp do wywołań API charakterystycznych dla AIR (w tym do wywołań Flash Playera) odbywa się za pośrednictwem właściwości `runtime`. Omówimy to bardziej szczegółowo w dalszej części książki.

UWAGA

Sprawdzenie właściwości `runtime` to prosty sposób ustalenia, czy aplikacja HTML/JavaScript działa w środowisku Adobe AIR. Aby sprawdzić tę właściwość, należy użyć następującego kodu:

```
if(window.runtime)
{
    // aplikacja działa w środowisku AIR
}
```

Skoro mamy już plik deskryptora aplikacji i główny plik HTML, możemy przetestować działanie aplikacji w środowisku uruchomieniowym.

Testowanie aplikacji

Choć niektóre środowiska IDE do pisania kodu HTML (na przykład Adobe Dreamweaver) obsługują bezpośrednie uruchamianie i testowanie aplikacji AIR, my skupimy się na narzędziu ADL wchodzącym w skład pakietu SDK. W ten sposób czytelnicy lepiej zrozumieją cały proces. Ponadto ADL zapewnia największą elastyczność, jeśli chodzi o integrację procesu tworzenia aplikacji z innymi środowiskami IDE, edytorami i obiegami pracy.

Uruchamianie aplikacji za pomocą ADL

Pierwszym etapem testowania aplikacji jest uruchomienie jej w środowisku AIR i upewnienie się, że:

- w pliku deskryptora aplikacji nie ma żadnych błędów;
- aplikacja uruchamia się;
- kod HTML jest wyświetlany prawidłowo;
- kod JavaScript działa w oczekiwany sposób.

Choć moglibyśmy spakować całą aplikację, a później ją zainstalować, byłoby to żmudne i utrudniło proces modyfikowania kodu oraz testowania nowych wersji. Na szczęście pakiet Adobe AIR SDK zawiera narzędzie wiersza poleceń o nazwie ADL, które umożliwia uruchomienie aplikacji AIR bez jej instalacji.

Aby przetestować aplikację, należy:

1. Otworzyć okno terminala (w Macu) albo okno konsoli (w Windows).

- Przejdź do katalogu, który zawiera pliki *AIRHelloWorld.html* i *AIRHelloWorld.xml*.
- Uruchomić program ADL za pomocą poniższego polecenia, przekazując nazwę pliku deskryptora aplikacji:

```
adl AIRHelloWorld.xml
```

Aplikacja powinna uruchomić się w standardowym oknie systemu operacyjnego (rysunek 2.2).



Rysunek 2.2. Aplikacja *AIRHelloWorld* uruchomiona za pomocą programu ADL w systemie Windows Vista

Jeśli aplikacja nie uruchomi się prawidłowo albo jeśli wystąpi błąd, należy sprawdzić, czy:

- pakiet SDK został skonfigurowany w taki sposób, że system może odnaleźć narzędzia ADL;
- polecenie ADL zostało wydane w katalogu, w którym znajduje się plik *AIRHelloWorld.xml*;
- plik deskryptora aplikacji zawiera poprawny składniowo kod XML;
- informacje w pliku deskryptora aplikacji są prawidłowe (szczególną uwagę należy zwrócić na atrybuty `application` i element `initialWindow`);

- pliki *AIRHelloWorld.html* i *AIRHelloWorld.xml* znajdują się w tym samym katalogu.

Po rozwiązaniu ewentualnych problemów i pomyślnym uruchomieniu aplikacji możemy zbadać, jak uzyskać informacje od aplikacji w czasie jej działania.

Przechwytywanie danych wyjściowych aplikacji

Kiedy aplikacja jest uruchamiana z wiersza poleceń za pośrednictwem ADL, można uzyskać jej dane wyjściowe i informacje diagnostyczne na kilka różnych sposobów.

Błędy JavaScriptu

Wszystkie błędy spowodowane wykonywaniem JavaScriptu w aplikacji uruchomionej za pośrednictwem ADL są przekazywane na standardowe wyjście ADL.

Zmodyfikujmy aplikację tak, aby w czasie wykonania generowała błąd JavaScriptu. Treść pliku *AIRHelloWorld.html* należy zmienić w następujący sposób:

```
<html>
<head>
  <title>AIRHelloWorld</title>

  <script>
    function init()
    {
      runtime2.trace("wywołano funkcję init");
    }
  </script>
</head>
<body onload="init()">
  <div align="center">Witaj, świecie</div>
</body>
</html>
```


Zmieniliśmy funkcję `init` w taki sposób, aby próbowała uzyskać dostęp do nieistniejącej właściwości o nazwie `runtime2`:

```
runtime2.trace("wywołano funkcję init");
```

Zapiszmy aplikację i wykonajmy ją za pośrednictwem ADL:

```
adl AIRHelloWorld.xml
```

Po uruchomieniu aplikacji w wierszu poleceń powinien pojawić się następujący komunikat o błędzie:

```
ReferenceError: Can't find variable: runtime2  
init at app:/AIRHelloWorld.html : 8  
init at app:/AIRHelloWorld.html : 8  
onload at app:/AIRHelloWorld.html : 13
```

Komunikat opisuje błąd (w tym przypadku polegający na tym, że nie odnaleziono zmiennej o nazwie `runtime2`) i pokazuje, w którym wierszu on wystąpił (8) oraz jaki był stos wywołań. Informacje te pomagają odnaleźć ewentualne błędy w aplikacji.

Zdarzają się również sytuacje, że aplikacja nie działa prawidłowo, ale nie zgłasza żadnych błędów. W takich przypadkach warto przechwycić informacje o stanie działającej aplikacji.

Adobe AIR zawiera funkcję, która umożliwia wysyłanie informacji z aplikacji na standardowe wyjście.

runtime.trace

Jak już wspomnieliśmy, Adobe AIR oferuje mechanizm do przesyłania łańcuchów z JavaScriptu do wiersza poleceń.

Funkcja `trace` właściwości `runtime` przyjmuje łańcuch, który jest wypisywany na standardowym wyjściu ADL. Oto przykład jej użycia:

```
runtime.trace("To zostanie wysłane na standardowe wyjście");
```

Przydaje się to do śledzenia stanu aplikacji bez przerywania działania kodu.

Kiedy prześlemy do funkcji `trace()` obiekt inny niż łańcuch, zostanie wywołana jego funkcja `toString()`. Obiekt `Object` JavaScriptu ma domyślną implementację `toString()`, ale niektóre klasy (takie jak `Array`) implementują bardziej kontekstowe funkcje `toString()`.

Oto przykład śledzenia tablicy, która zawiera różne typy danych:

```
var a = ["a", 1, {foo:"bar"}];
runtime.trace(a);
```

Kod ten spowoduje wyświetlenie poniższych danych w wierszu polecenia ADL:

```
a,1,[object Object]
```

Oczywiście, można implementować własne metody `toString()` w niestandardowych klasach JavaScriptu albo przesłaniać funkcje `toString()` istniejących klas, aby uzyskać wyniki bardziej specyficzne dla danej klasy.

Pakowanie i wdrażanie aplikacji AIR

Wiemy już, jak budować, testować i debugować aplikację AIR, więc jesteśmy gotowi do utworzenia pliku AIR, który umożliwi wdrażanie i dystrybucję aplikacji.

Co znajduje się w pliku AIR?

Plik AIR to pakiet w formacie ZIP, który służy do dystrybuowania aplikacji AIR. Zawiera wszystkie pliki niezbędne do zainstalowania i uruchomienia aplikacji AIR, a środowisko Adobe AIR używa go do utworzenia i zainstalowania aplikacji w systemie użytkownika.

Plik AIR jest tworzony przez narzędzie ADT z pakietu AIR SDK i używany do rozpowszechniania aplikacji wśród innych użytkowników.

UWAGA

Instalacja pliku AIR wymaga, aby w systemie użytkownika było już zainstalowane środowisko Adobe AIR.

Plik AIR zawiera przynajmniej dwa elementy: plik deskryptora aplikacji oraz główny plik aplikacji. Można jednak dołączyć do niego inne pliki, ikony, katalogi i zasoby, które zostaną zainstalowane wraz z aplikacją. Pliki te będą dostępne dla uruchomionej aplikacji.

Potrzebny też będzie certyfikat do cyfrowego podpisania aplikacji.

Podpisywanie plików AIR

Adobe AIR wymaga, aby wszystkie aplikacje AIR były podpisane cyfrowo. Można to zrobić na dwa sposoby.

Podpisywanie za pomocą samodzielnie podpisanego certyfikatu

Programiści mogą użyć ADT, aby cyfrowo podpisać plik AIR za pomocą samodzielnie podpisanego certyfikatu. W tym celu należy wygenerować samodzielnie podpisany certyfikat, a następnie wykorzystać go do podpisania pliku AIR. Samodzielnie podpisane pliki AIR nie zapewniają bezpieczeństwa i nie pozwalają sprawdzić, czy autor pliku rzeczywiście jest tym, za kogo się podaje. Podczas instalowania aplikacji środowisko Adobe AIR ostrzeże użytkownika, że wydawca aplikacji nie może być zweryfikowany.

UWAGA

Nie można zmienić typu certyfikatu podczas aktualizowania aplikacji.

Pliki AIR podpisane za pomocą samodzielnie podpisanych certyfikatów przydają się głównie podczas tworzenia aplikacji. Jeśli aplikacja ma być rozpowszechniana publicznie, trzeba podpisać ją za pomocą certyfikatu wystawionego przez zaufany i dobrze znany urząd certyfikacji (CA).

Podpisywanie za pomocą certyfikatu wystawionego przez CA

ADT obsługuje również podpisywanie aplikacji za pomocą certyfikatu wystawionego przez zaufany urząd CA. Dzięki temu środowisko Adobe AIR może zweryfikować wydawcę aplikacji i poinformować o tym użytkownika w oknie instalacyjnym.

UWAGA

Więcej informacji o podpisywaniu certyfikatów AIR, w tym listę urzędów CA wystawiających certyfikaty zgodne z Adobe AIR, można znaleźć pod adresem: http://www.adobe.com/go/learn_air_html.

Oba typy certyfikatów — podpisane samodzielnie i wystawione przez CA — pozwalają środowisku Adobe AIR sprawdzić, czy plik AIR nie został przez kogoś zmodyfikowany.

Ponieważ nie będziemy rozpowszechniać tworzonej właśnie aplikacji, podpiszemy nasz plik AIR za pomocą samodzielnie podpisanego certyfikatu.

Tworzenie pliku AIR za pomocą ADT

Tworzenie samodzielnie podpisanego pliku AIR składa się z dwóch etapów:

1. Należy użyć ADT do utworzenia samodzielnie podpisanego certyfikatu.

2. Korzystając z ADT, należy utworzyć plik AIR podpisany za pomocą samodzielnie podpisanego certyfikatu.

Generowanie samodzielnie podpisanego certyfikatu

Przed podpisaniem pliku AIR za pomocą samodzielnie podpisanego certyfikatu musimy wygenerować certyfikat.

Aby wygenerować samodzielnie podpisany certyfikat za pomocą ADT, należy użyć następujących opcji wiersza polecenia:

```
adt -certificate -cn NAZWA TYPKLUCZA NAZWAPLIKU HASŁO
```

Opcje te są wyjaśnione w tabeli 2.4.

Tabela 2.4. Opcje podpisywania w programie ADT

Opcja wiersza polecenia	Opis
NAZWA	Nazwa nowego certyfikatu.
TYPKLUCZA	Typ klucza użytego w certyfikacie, 1024-RSA albo 2048-RSA.
NAZWAPLIKU	Nazwa pliku, w którym zostanie zapisany certyfikat.
HASŁO	Hasło certyfikatu.

Aby wygenerować nowy samodzielnie podpisany certyfikat, należy wykonać poniższe czynności:

1. Otworzyć okno terminala (Mac OS X) albo konsoli (Windows).
2. Przejść do katalogu, który zawiera pliki *AIRHelloWorld.html* i *AIRHelloWorld.xml*.
3. Wydać poniższe polecenie:

```
adt -certificate -cn foo 1024-RSA test_cert.p12 mypass
```

W tym przykładzie nadajemy certyfikatowi nazwę *foo* i hasło *mypass*.

Polecenie to wygeneruje samodzielnie podpisany certyfikat i zapisze go w pliku o nazwie *test_cert.p12*.

UWAGA

Tego samego samodzielnie podpisanego certyfikatu można użyć do podpisania wielu plików AIR.

W tym momencie powinniśmy mieć plik o nazwie *test_cert.p12* w tym samym katalogu co pliki aplikacji. Teraz możemy użyć tego certyfikatu do cyfrowego podpisania pliku AIR.

Generowanie pliku AIR

Pliki AIR tworzy się za pomocą narzędzia ADT dołączonego do pakietu Adobe AIR SDK. Używa się go w następujący sposób:

```
adt -package OPCJEPODPISU NAZWAPLIKUAIR DOŁĄCZANEPLIKI
```

Aby utworzyć plik AIR podpisany za pomocą samodzielnie podpisanego certyfikatu, należy wykonać poniższe czynności:

1. Otworzyć okno terminala (Mac OS X) albo konsoli (Windows).
2. Przejść do katalogu, który zawiera pliki *AIRHelloWorld.html* i *AIRHelloWorld.xml*.
3. Wydać poniższe polecenie:

```
adt -package -storetype pkcs12 -keystore test_cert.p12  
  AIRHelloWorld.air AIRHelloWorld.xml AIRHelloWorld.html
```

4. Po wydaniu polecenia powinno pojawić się pytanie o hasło do certyfikatu. Należy wprowadzić hasło, w tym przypadku **mypass**.

UWAGA

Podczas podpisywania pliku AIR narzędzie ADT próbuje skontaktować się z internetowym serwerem czasu, aby oznaczyć plik bieżącą datą i godziną. Jeśli połączenie nie powiedzie się, zostanie wyświetlony następujący komunikat o błędzie:

```
Could not generate timestamp
```

Podczas samodzielnego podpisywania pliku AIR można ominąć ten błąd, nakazując narzędziu ADT, aby nie oznaczało pliku bieżącą datą i godziną; w tym celu należy dodać do wiersza polecenia następującą opcję:

```
-tsa none
```

W tym przypadku całe polecenie miałyby postać:

```
adt -package -storetype pkcs12 -keystore  
↳test_cert.p12 -tsa none AIRHelloWorld.air  
↳AIRHelloWorld.xml AIRHelloWorld.html
```

Powyższe polecenie powinno utworzyć plik o nazwie *AIRHelloWorld.air* w tym samym katalogu, w którym znajdują się pliki aplikacji. Jeśli plik nie zostanie utworzony albo pojawią się jakieś błędy, należy sprawdzić, czy:

- pakiet SDK jest skonfigurowany prawidłowo, a narzędzie ADT znajduje się na ścieżce systemowej;
- polecenie ADT zostało wydane w katalogu, w którym znajduje się plik *AIRHelloWorld.xml*;
- plik deskryptora aplikacji zawiera poprawny składniowo kod XML;
- informacje w pliku deskryptora aplikacji są prawidłowe (szczególną uwagę należy zwrócić na atrybuty `application` i `element content`);
- pliki *AIRHelloWorld.html*, *test_cert.p12* oraz *AIRHelloWorld.xml* znajdują się w tym samym katalogu;

- podano to samo hasło, co podczas generowania pliku certyfikatu.

Testowanie i instalowanie pliku AIR

Po utworzeniu pliku AIR pozostaje nam tylko przetestować go i sprawdzić, czy instaluje się prawidłowo.

Aby przetestować plik AIR, trzeba zainstalować go w systemie, a następnie uruchomić. W tym celu należy:

1. W programie Eksplorator Windows (Windows) albo Finder (Mac OS X) przejść do katalogu, który zawiera plik AIR.
2. Kliknąć dwukrotnie plik AIR.
3. Postępować według wskazówek wyświetlonych w oknie instalacji.
4. Na ostatnim ekranie okna instalacji zaznaczyć pole wyboru *Start Application after installation*.

Aplikacja powinna się uruchomić. Jeśli się nie uruchomi albo pojawi się komunikat o błędzie, należy sprawdzić, czy:

- w systemie została zainstalowana wersja 1.0 Adobe AIR;
- podczas tworzenia pliku AIR za pomocą ADT nie wystąpiły żadne błędy;
- odinstalowano wstępne wersje Adobe AIR.

Jeśli aplikacja została zainstalowana i działa prawidłowo, można uruchomić ją ponownie przez kliknięcie jej ikony. Domyślna lokalizacja skrótu zależy od systemu operacyjnego. W Macu skrót jest tworzony w folderze */Applications*. W Windows można go znaleźć w menu *Start/Programy/<NAZWA APLIKACJI>*.

Wdrażanie pliku AIR

Po pomyślnym utworzeniu i spakowaniu aplikacji AIR można przystąpić do jej rozpowszechniania. Aplikacje AIR można dystrybuować w sieci WWW albo bezpośrednio na płytach CD-ROM lub innych nośnikach.

UWAGA

Obszerne informacje o rozpowszechnianiu aplikacji Adobe AIR przez sieć WWW można znaleźć w dokumentacji Adobe AIR oraz w samouczku pod adresem: http://www.adobe.com/devnet/air/articles/air_badge_install.html.

Ustawianie typu MIME

Jeśli pliki AIR mają być udostępniane do pobrania przez serwer WWW, należy zwrócić uwagę, aby w serwerze ustawiony był prawidłowy typ MIME. Jeśli typ MIME nie jest ustawiony poprawnie, przeglądarki mogą potraktować plik AIR jak plik ZIP (przy okazji zmieniając jego nazwę) albo wyświetlić bajty pliku w oknie przeglądarki zamiast pobrać aplikację do systemu użytkownika.

Prawidłowy typ MIME dla pliku AIR to:

```
application/vnd.adobe.air-application-installer-package+zip
```

Aby na przykład ustawić typ MIME w serwerze Apache, należy dodać następujący wiersz do pliku konfiguracyjnego serwera:

```
AddType application/vnd.adobe.air-application-  
installerpackage+zip .air
```

Informacje dotyczące ustawiania typów MIME w innych serwerach można znaleźć w ich dokumentacji.

W tym momencie czytelnicy wiedzą już, jak tworzyć, testować i wdrażać proste aplikacje AIR. Teraz pora powiedzieć więcej o wywołaniach API AIR oraz o budowaniu bardziej funkcjonalnych i zaawansowanych aplikacji.