

Using Adobe AIR for Online Help

- ❖ Introduction
- ❖ What is Adobe AIR?
- ❖ What is AIR Help?
- ❖ Benefits of AIR Help
- ❖ Adobe RoboHelp AIR Output
- ❖ MadCap Flare AIR Output
- ❖ Comparison: RoboHelp vs. Flare
- ❖ Custom AIR Help Options
- ❖ Custom AIR Help Example
- ❖ AIR Application Development
- ❖ HTML-Based AIR Project
- ❖ MXML-Based AIR Project
- ❖ AIR Help Possibilities
- ❖ Wrap Up

Introduction

- ❖ Scott Prentice, President of Leximation, Inc.
- ❖ Specializing in FrameMaker plugin development as well as structured FrameMaker conversions, consulting, and development. FrameMaker user/developer since 1991.
- ❖ Developed DITA-FMx, a FrameMaker plugin for efficient DITA authoring and publishing.
- ❖ Developer of custom Help systems and creative/functional web applications.
- ❖ Coined the term “AIR Help” in 2007 after learning about Adobe’s new AIR technology.
- ❖ Interested in creating innovative ways to provide user assistance that is actually used.

What is Adobe AIR?

- ❖ Adobe® AIR™ (Adobe Integrated Runtime) is a cross-platform development environment that lets developers create native desktop applications using web technologies like HTML and Flash.
- ❖ Installation of an AIR application requires the prior installation of the AIR runtime.
- ❖ The same “.air” file can be installed on Windows, Mac, and Linux systems.
- ❖ AIR is based on Open Source technologies - Flex (ActionScript, MXML), WebKit, SQLite, Flash.
- ❖ Everything that you need to develop and create an AIR application is available as a free download from adobe.com (Flex and AIR SDKs).
- ❖ AIR supports the following 16 languages: Chinese (Traditional and Simplified), Czech, Dutch, English, French, German, Italian, Japanese, Korean, Polish, Portuguese, Russian, Spanish, Swedish, and Turkish.

What is AIR Help?

- ❖ “AIR Help” is a term that describes a user assistance application that has been developed using the Adobe AIR development platform.
- ❖ Because the AIR technology is so flexible, Help systems developed using AIR may provide very different features.
- ❖ Two commercially available tools for exporting an AIR Help file:
 - ◆ Adobe’s *RoboHelp 8* – Adobe AIR-based Help
 - ◆ MadCap Software’s *Flare 4* or *Flare 5* – WebHelp AIR
- ❖ Design and develop your own AIR Help system using HTML/Ajax/JavaScript or Flex/Flash technologies. Complexity ranges from “wrapping up” an existing HTML-based Help system to a highly developed custom user assistance application.
- ❖ Sample AIR Help source files are available from Leximation. A prototype AIR Help plugin for the DITA Open Toolkit is currently available.

Benefits of AIR Help

- ❖ Cross-platform
- ❖ Single deliverable for a Help system (like a CHM file)
- ❖ Embedded browser component eliminates the need to design for multiple browsers
- ❖ Highly customizable, limited only by your imagination
- ❖ Strong connection to the web, easy integration of web and local content
- ❖ Easily embed Flash and PDF content
- ❖ Can read/write data to the local file system
- ❖ Security, each AIR file must be digitally signed by the publisher
- ❖ Open source development technologies

Adobe RoboHelp AIR Output

- ❖ Provides typical Help system features such as Contents, Index, Search, Glossary, Favorites, Print, and Forward/Back.
- ❖ Supports context sensitive help calls.
- ❖ Variety of skins/themes to customize the look and feel
- ❖ Comments added to topics can be shared between users (a great way to perform reviews of online content)
- ❖ Mini TOC links to sub heads in each topic
- ❖ Include context with search results (like Google)
- ❖ Auto-update for online updates of Help system
- ❖ Include additional web-based content on Resource tabs and via RSS
- ❖ Install a single file help package (as opposed to loose HTML files)
- ❖ The *RoboHelp Packager for AIR* reads RoboHelp 6 or 7 WebHelp files and transforms them into an Adobe AIR-based Help application.

Adobe RoboHelp AIR Output (cont.)

The screenshot displays the Adobe RoboHelp AIR interface for the 'DITA-FMx (RH)' project. The main content area shows the 'Structure Application Requirements' page, which provides details on the element definitions used with DITA-FMx. The page includes a search bar, a navigation pane on the left, and a footer with 'Powered by Leximation' and 'About Preferences' options.

Structure Application Requirements

Provides details on the element definitions used with DITA-FMx.

The easiest way to get up and running with DITA in Frame is to use the default structure applications provided with DITA-FMx; they will "just work" right out of the box. Once you're comfortable with things and want to make some adjustments to fit your house style or want to specialize, you can either clone the apps provided, use some DITA apps provided by someone else, or make your own from scratch. If you want to customize the apps provided with DITA-FMx, refer to the topic [Customizing the Default Structure Applications](#). If you want to use other applications or want to make your own, just be sure that they adhere to the requirements expected by DITA-FMx. In order to implement certain DITA features in DITA-FMx, the applications must be set up in a very specific way.

The information in the following topics describes the requirements that need to exist in a DITA structure application so that it functions properly with DITA-FMx. All of the modifications described affect the structure application files only (EDD, template, rules file), no modifications need to be made to the DITA DTD files. The items are grouped by feature, some of these features require the addition of multiple elements or structure application modifications. If you don't want to make use of a given feature, you can skip the modifications described in that section. There are many other rules and EDD modifications that are needed for proper use of DITA files in FrameMaker, these are not discussed; only those features that are required or used by DITA-FMx are documented at this time.

The easiest way to set up these features in your custom application is to copy and paste from the default DITA-FMx applications.

fm-xref

The fm-xref element is used to create a FrameMaker-style cross-reference used inline as a standard DITA xref element would be used. On file save, this converts to a

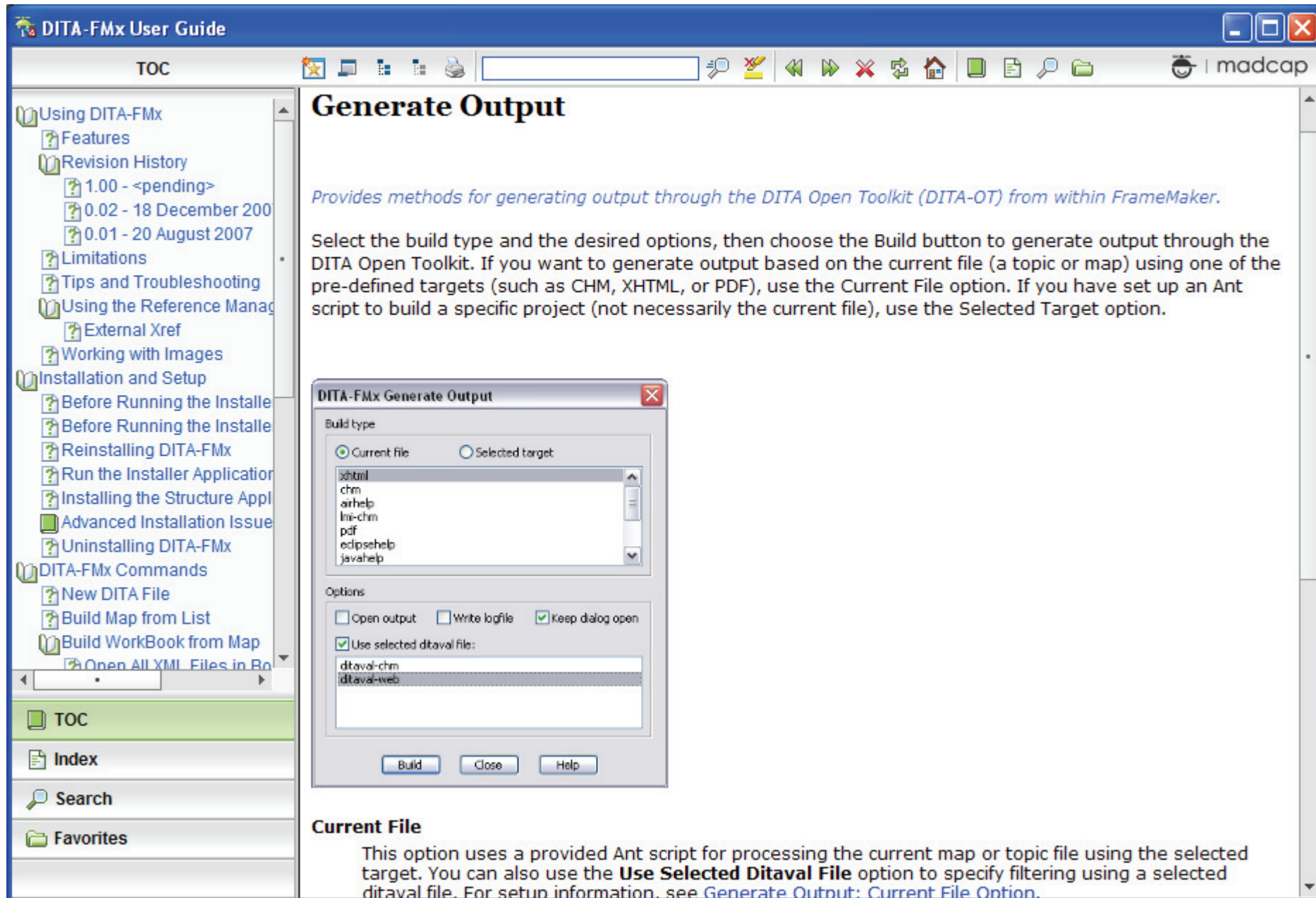
Powered by Leximation

About Preferences 100 %

MadCap Flare AIR Output

- ❖ Flare's AIR output is a “wrapped” WebHelp system. All of the features are HTML-based and not defined with programmatic Flex/ActionScript controls.
- ❖ Provides typical Help system features such as Contents, Index, Search, Glossary, Favorites, and Forward/Back.
- ❖ Other features can easily be added by modifying the underlying WebHelp system.

MadCap Flare AIR Output (cont.)



The screenshot shows the DITA-FMx User Guide interface. On the left is a Table of Contents (TOC) with categories like 'Using DITA-FMx', 'Installation and Setup', and 'DITA-FMx Commands'. The main content area is titled 'Generate Output' and contains the following text:

Generate Output

Provides methods for generating output through the DITA Open Toolkit (DITA-OT) from within FrameMaker.

Select the build type and the desired options, then choose the Build button to generate output through the DITA Open Toolkit. If you want to generate output based on the current file (a topic or map) using one of the pre-defined targets (such as CHM, XHTML, or PDF), use the Current File option. If you have set up an Ant script to build a specific project (not necessarily the current file), use the Selected Target option.

A dialog box titled 'DITA-FMx Generate Output' is overlaid on the main content. It has two radio buttons: 'Current file' (selected) and 'Selected target'. Below these is a list box containing: xhtml, chm, airhelp, ini-chm, pdf, eclipsehelp, and javahelp. Under the 'Options' section, there are checkboxes for 'Open output', 'Write logfile', 'Keep dialog open', and 'Use selected ditaval file:'. The 'Use selected ditaval file:' checkbox is checked, and a list box below it contains 'ditaval-chm' and 'ditaval-web'. At the bottom of the dialog are 'Build', 'Close', and 'Help' buttons.

Current File

This option uses a provided Ant script for processing the current map or topic file using the selected target. You can also use the **Use Selected Ditaval File** option to specify filtering using a selected ditaval file. For setup information, see [Generate Output: Current File Option](#).

Comparison: RoboHelp vs. Flare

Specific features available with each AIR output option.

AIR Help Feature	RoboHelp	Flare
Natively exports AIR from authoring tool	✓	✓
Context sensitive Help calls	✓	
Auto-update capability	✓	
Allows user comments/review	✓	✓ (a)
Dynamic mini TOC	✓	
Search results with context	✓	
Multiple resource tabs for external websites	✓	✓ (b)
Ability to customize the file icon	✓	
Single file Help package	✓	
Languages supported	16	16
Persistent window size and position		

- a. Available through the purchase of the MadCap Feedback Server application
- b. Could be added as a topic in the main WebHelp project

Custom AIR Help Options

- ❖ If you're willing to do some coding and development, your options for a Help delivery system are essentially unlimited.
- ❖ It is very easy to wrap an existing browser-based Help system (generated by any Help authoring tool) in AIR. This results in the same functionality as if it were not wrapped in AIR, but gives you a cross-platform, single file deliverable.
- ❖ A custom user assistance application can look like a “traditional” Help system (navigation on the left, content on the right), or it can provide additional features or introduce a completely different paradigm.

Custom AIR Help Example

- ❖ The Leximation DITA to AIR Help plugin for the DITA Open Toolkit is available free of charge.
- ❖ This provides the basic functionality that you'd expect in a Help system, and is intended to be used as the starting point for developing your own custom AIR Help system.
- ❖ Even if you're not using DITA, you can make use of this AIR application code for your own custom AIR Help system.
- ❖ Features provided:
 - ◆ Standard navigation tabs for Contents, Index, and remote/local full text search with word highlighting
 - ◆ Next, Previous, Forward, Back, and Home buttons
 - ◆ Context sensitivity
 - ◆ Persistent window size and position

Custom AIR Help (cont.)

DITA-FMx User Guide

ditafmx_using.html

Search Index Contents

- Using DITA-FMx
 - Features
 - Revision History
 - Limitations
 - Tips and Troubleshooting
 - Using the Reference Manager
 - Working with Images
 - Filtering Content
- Installation and Setup
- DITA-FMx Commands
- Extending DITA-FMx

Using DITA-FMx

Documentation last updated: 22 February 2009

Updated for plugin versions: authoring support client v.1.1.01; import/export client v.1.1.01.

Attention: This documentation is under development .. use with care.

Features
Provides DITA map and topic authoring commands as well as enhanced DITA features.

Revision History
Describes the changes between versions of DITA-FMx.

Limitations
Known limitations in the current version of DITA-FMx.

Tips and Troubleshooting
Tips for making the most efficient use of DITA-FMx.

Using the Reference Manager
Lets you select a conref, xref, or link target by specifying the file, element type, and element.

Working with Images
Information and tips regarding the image handling features.

Filtering Content
Tips for filtering and conditionalizing content in topic files or maps.

Related information
[Features](#)
[Installation and Setup](#)
[DITA-FMx Commands](#)

<< Previous Next >>

AIR Application Development

- ❖ Development of an AIR application is typically done with Adobe's Flex Builder, although this can be done in other development environments or with a simple text editor.
- ❖ The interface of a typical AIR application is laid out with MXML, an XML format that provides for the ability to arrange interface components.
- ❖ ActionScript is used to handle events and dynamic interaction with the interface components. ActionScript is an object oriented programming language similar to JavaScript.
- ❖ An HTML browser component (WebKit) can be added to the interface to render HTML within the application.
- ❖ Your user assistance application can be designed to take on the operating system's look and functionality for the window elements, or it can be customized. You can even create windows that are a shape other than a rectangle.

HTML-Based AIR Project

❖ Application descriptor file (*hello-app.xml*)

```
<?xml version="1.0" encoding="UTF-8"?>
<application xmlns="http://ns.adobe.com/air/application/1.5.2">
  <id>com.leximation.ebstc.hello</id>
  <filename>hello</filename>
  <version>0.01</version>
  <name>Adobe AIR Basics</name>
  <initialWindow>
    <content>hello.html</content>
    <visible>true</visible>
  </initialWindow>
</application>
```

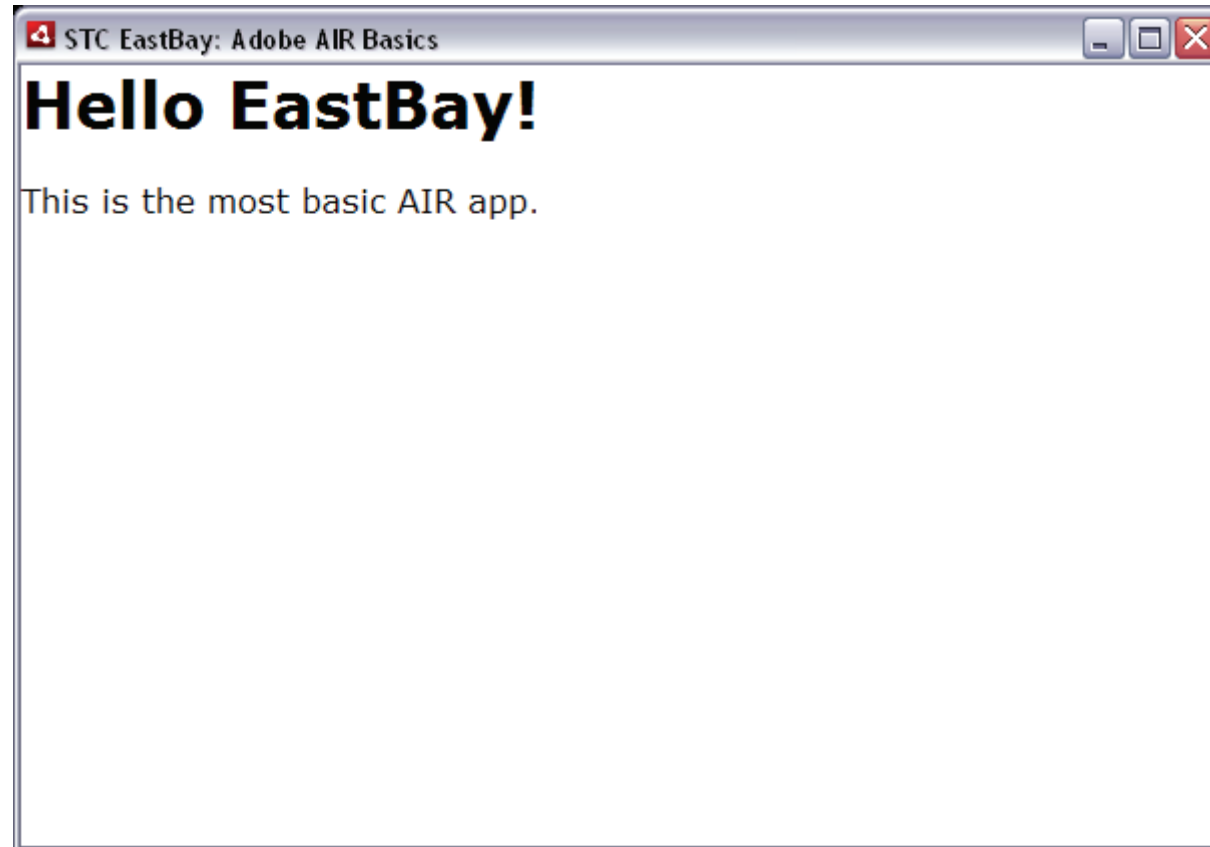
❖ Root content file (*hello.html*)

```
<html>
<head><title>STC EastBay: Adobe AIR Basics</title></head>
<body>
<h1>Hello EastBay!</h1>
<p>This is the most basic AIR app.</p>
</body>
</html>
```

❖ Test using ADL.

```
"C:\Tools\Flex\flex_sdk_3.4\bin\adl.exe" hello-app.xml
```

HTML-Based AIR Project (cont.)



MXML-Based AIR Project

❖ Application descriptor file (*helloAgain-app.xml*)

```
<?xml version="1.0" encoding="UTF-8"?>
<application xmlns="http://ns.adobe.com/air/application/1.5.2">
  <id>com.leximation.ebstc.helloAgain</id>
  <filename>helloAgain</filename>
  <version>0.01</version>
  <name>Adobe AIR Basics</name>
  <initialWindow>
    <content>helloAgain.swf</content>
    <visible>true</visible>
  </initialWindow>
</application>
```

❖ Root content file (*helloAgain.mxml*)

```
<?xml version="1.0" encoding="utf-8"?>
<mx:WindowedApplication xmlns:mx="http://www.adobe.com/2006/mxml"
  layout="absolute">
  <mx:Script>
    <![CDATA[
      import mx.collections.*;
      private function initHtmlControl():void {
        htmlControl.location='help/hello.html'
      }
    ]]>
  </mx:Script>
```

... continued on next page

MXML-Based AIR Project (cont.)

❖ Root content file continued (*helloAgain.mxml*)

... continued from previous page

```
<mx:VBox width="100%" height="100%">
  <mx:HBox width="100%" paddingTop="3" paddingRight="3"
paddingLeft="3">
    <mx:TextInput id="textInput" width="100%"
      editable="false" backgroundColor="#eeeeee"/>
    <mx:Button label="Home" click="initHtmlControl()"/>
    <mx:Button label="Google"
      click="htmlControl.location='http://www.google.com'"/>
    <mx:Button label="STC EastBay"
      click="htmlControl.location='http://www.ebstc.org'"/>
  </mx:HBox>
  <mx:HTML id="htmlControl" paddingLeft="10" paddingBottom="10"
paddingRight="10" paddingTop="10" height="100%" width="100%"
borderStyle="none" bottom="0" horizontalCenter="0"
creationComplete="initHtmlControl()"
complete="textInput.text=htmlControl.location"/>
</mx:VBox>
</mx:WindowedApplication>
```

❖ Compile SWF using AMXMLC.

```
"C:\Tools\Flex\flex_sdk_3.4\bin\amxmlc.bat" helloAgain.mxml
```

MXML-Based AIR Project (cont.)

❖ Test using ADL.

```
"C:\Tools\Flex\flex_sdk_3.4\bin\adl.exe" helloAgain-app.xml
```

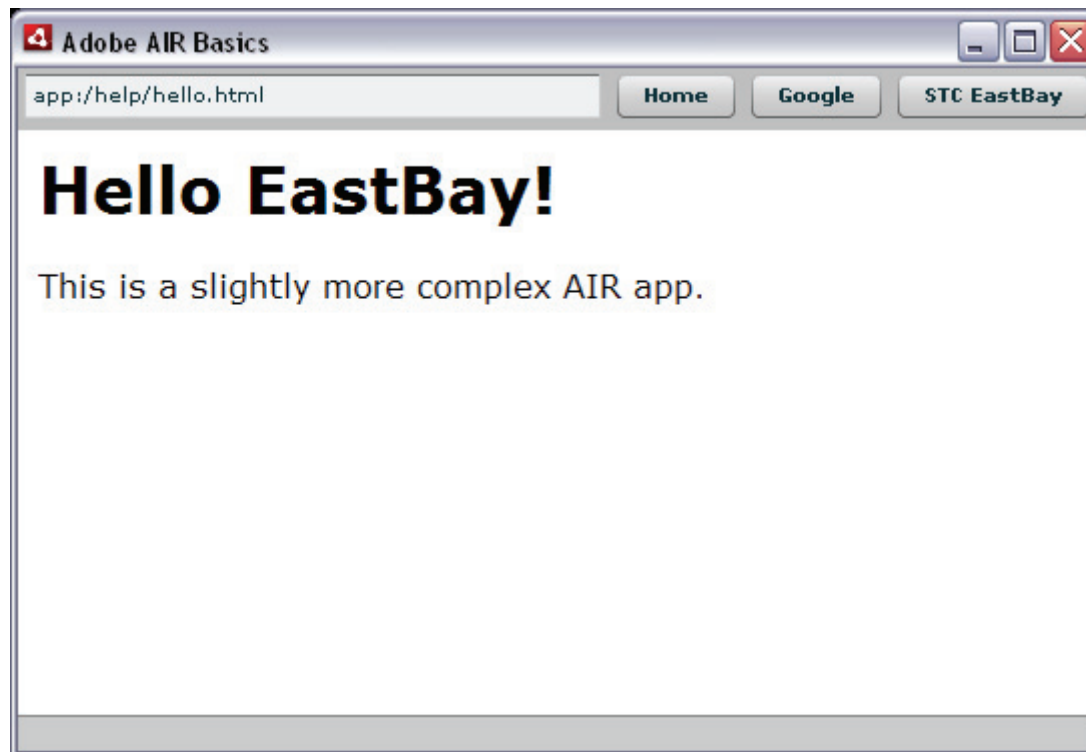
❖ Create self-signed certificate using ADT.

```
"C:\Tools\Flex\flex_sdk_3.4\bin\adt" -certificate -cn mySelfSignedCert  
1024-RSA myCert.pfx secretPassword
```

❖ Create the AIR package file using ADT.

```
"C:\Tools\Flex\flex_sdk_3.4\bin\adt" -package -storetype pkcs12 -keystore  
myCert.pfx -storepass secretPassword helloAgain.air helloAgain-app.xml  
helloAgain.swf help/*
```

MXML-Based AIR Project (cont.)



AIR Help Possibilities

Aside from “traditional” Help system features, some possibilities are:

- ❖ Combined remote/local search
- ❖ User comments visible to all users of the Help system
- ❖ A chat interface that lets users ask questions of other users of the Help system
- ❖ Ability for users and administrators to add new topics to the Help system that describe their best practices for using the tool
- ❖ Ability for users to customize the navigation or other UI elements
- ❖ Dynamically download content from the web based on the user’s needs or user “type”
- ❖ Dynamically update content as it becomes available
- ❖ Much, much more...

Wrap Up

Resources:

- ❖ <http://www.adobe.com/products/air/>
- ❖ <http://www.adobe.com/products/robohelp/>
- ❖ <http://www.adobe.com/products/robohelp/robohelp-packager-for-air/>
- ❖ <http://madcapsoftware.com/products/flare/>
- ❖ <http://www.leximation.com/airhelp/>
- ❖ http://tech.groups.yahoo.com/group/AIR_Help/

Questions?

- ❖ Scott Prentice <sp AT leximation DOT com>
- ❖ Leximation - <http://www.leximation.com>