Chapter 6

Go Fly

A Real-World Apollo Application

This Apollo application, written in Flex (the serverside is a ColdFusion component, or CFC), presents a simple user interface that accepts an airline code and a flight number. Clicking the Search button calls a web service that communicates with a third-party, web-based infrastructure that provides close to realtime flight-tracking information (updates are broadcast at approximately six-minute intervals). Knowing the six-minute time lag between flight information updates, the Apollo application uses the Flex `Timer` class to poll the web service every six minutes. Polling that is more frequent would be a waste of resources. By leveraging Flex Data Services (now called Live Cycle Data Services), the Apollo application could detect a real-time update (push technology). Figure 6.1 shows the Apollo UI, and Figure 6.2 displays the flight details.

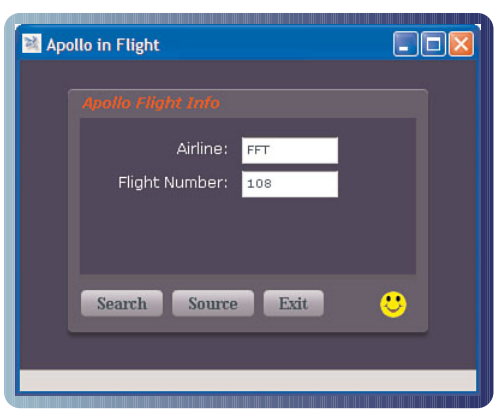

FIGURE 6.1
Apollo Go Fly sample UI.

Building the application includes working with some common techniques used in Flex development. These techniques include using a timer to periodically do something, using a component to display the flight details, using form validation to ensure that the airline and flight number are entered before the web service is called, using a `[Bindable]` metadata tag to bind the flight data to a persistent string variable, and using handlers to perform some unit of work. For this application, the Apollo `HTML` class and `NativeWindow` class are used to display the flight information in an HTML control inside an OS native window.

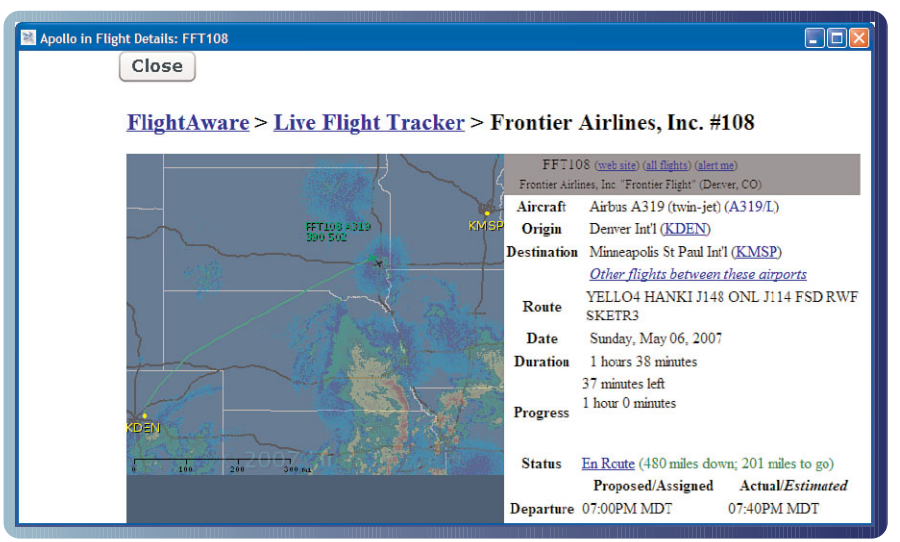

FIGURE 6.2
Apollo Go Fly flight details.

The source code for this Apollo application is available at

http://labs.insideflex.com/apollo-training/gofly/bin/srcview/index.html

The AIR file is available at

http://webmxml.no-ip.info/apollo-training/gofly/gofly.air