Completing the Service Agent

Over the course of the past several chapters you have been progressively applying new concepts to build the MovieBrowser sample application. In Chapter 14, you created a service agent that provided hard-coded sample data to the ViewModel. In this chapter, you will utilize the OData catalog provided by Netflix to create a service agent for the application that retrieves its data from a live service running in the cloud. In creating this service agent, you will also apply what you learned about asynchronous programming in Chapter 13.

Asynchronous Programming and the UI Thread

While the tools, frameworks, and technologies surrounding Windows programming have changed considerably over the years, the most fundamental concept of managing an event-based programming environment has remained unchanged. Windows applications utilize a procedure known as a message pump, which is essentially an endless loop that polls an input queue for messages from the operating system that are generated by user interaction such as input on the mouse and keyboard. These messages are then sent, or dispatched, to the application for processing as events. Once the application has completed processing the message, the cycle begins again. Figure 16-1 shows a conceptual representation of the message pump.
Using a message pump is an effective and simple way to manage an event-driven environment, but it has one inherent weakness: the code written in event handlers executes inline with the message loop, as illustrated with an event handler calling a long-running service method in the sequence diagram in Figure 16-2. While the object responsible for invoking events, known as the Dispatcher, waits for the method call on the service to return, additional messages are being placed in the queue and waiting to be processed. This leaves the end user at best feeling like the application is sluggish and at worst seeing portions of the window that can’t redraw or receiving a “not responding” message from Windows.

**Figure 16-1. Message pump**

**Figure 16-2. Event with synchronous service call**