Name: _____ Date:_____

Lab 5 – Windows Communication Foundation

Objectives

Part 1 – Functional Requirements

Reflection

Background/Scenario

In Lab 4, we have explored the use of accessing web services and APIs to access content on the web to dynamically change the functionality of the application. In this lab you will be using Windows Communication Foundation to create your own service host for client applications to access data from.

Windows Communication Foundation or WCF is a very easy to use framework that facilitates communication between applications in a service-oriented architecture. It abstracts the complexities of writing the networking aspect of the code so developers can focus on the business logic of the application.

In Lab 2, you wrote an application that presents the user with various font-related information and provided various user interface elements to allow the user to make comments or favorite fonts. In Lab 3, you extended the functionality using serialization to persist the data entered into the application. In this lab, you will once again extend the functionality of the font application and provide a service to host and store your font data. The clients would then be able to connect to the host, retrieve the data, and display it back to the user.

Required Resources

- Visual Studio
- Lab 3 Completed

Part 1: Functional Requirements

The end goal of this lab is to provide “cloud” functionality to the font application you originally wrote in Lab 2. In addition to adding client functionality in the existing application, you will also write a separate host application which will be the server to host the data.

General Requirements

- Use WCF to create a service host to store font information (favorite and comments).
  - Use TCP binding for the endpoint.
  - Use the app.config file to store WCF configuration information.
  - Expose metadata (create a MEX endpoint) on the service so the client can add a service reference to the host from Visual Studio.
  - Automatically load and store the server’s font information on application load and exit. Use whatever serialization format you wish.
- Update Lab 3 to connect to the WCF host with the following features:
  - Download font data and display the data to the UI
  - Upload current font data displayed in the UI to the server.

    - Connect to the WCF service by adding a service reference in Visual Studio.
- The host should be able to serve multiple clients simultaneously.
    - When debugging, launch the host, then several instances of the client. If you update the font information on one client instance and synchronize the changes to the server, when you refresh all the other clients, all the data should match.

User Interface Requirements

- Host Application
    - Must be written in WPF.
    - Provide user interface elements to show if the WCF service is running.
    - Provide user interface elements to start and stop the service.
- Client Application
    - Provide user interface elements to upload the current font information to the server.
    - Provide user interface elements to download the font information from the server. Once downloaded, the font information should immediately display.
    - Use exception handling to catch WCF communication errors.
        - Display in the user interface if WCF communication failed with a useful error message.

Extra Credit (Optional)

- Using a secondary endpoint address, expose a JSON service (http endpoint) on the host that returns a JSON array of the font data the server is hosting. (The host application should be able to serve both endpoints simultaneously.)

Reflection

1. In this lab, you have used TCP binding for the client and host applications to communicate with. What are some advantages or disadvantages of using any of the other bindings that WCF supports for this lab? Would it be better to use another binding instead?

basic binding and web binding. The other bindings won't be as easy to use because it requires more code to encapsulate all the functionality that I want to support from Lab 3. Disadvantages are: longer processing, more code. It would not be better (in this case) to use other bindings.