Self-Hosting WCF Services

Windows Communication Foundation (WCF) is a framework for building service-oriented applications. Using WCF, you can send data as asynchronous messages from one service endpoint to another. A service endpoint can be part of a continuously available service hosted by IIS, or it can be a service hosted in an application.

In order for clients to call a WCF service, you must host the service in a runtime environment. The role of the host application is to start and stop the service, listen for requests from clients, direct those requests to the service and send responses back to the clients.

To host a service, the host application uses the ServiceHost class in the System.ServiceModel namespace. The application uses this class to instantiate the service, configure endpoints, apply security and start listeners that handle requests from clients.

One of the benefits of hosting a WCF service using IIS or WAS is that each automatically creates an instance of the ServiceHost class when a client calls the service. You simply need to specify what service the host should activate. You do this in a .svc file by setting the Service property of the ServiceHost directive. The endpoints are defined in the Web.config file.

In order to host and start the service, you would need to follow steps below:

// Create a ServiceHost for the WindowsService type and provide the base address

serviceHost = new ServiceHost(typeof(DemoService));

// Open the ServiceHostBase to create listeners and start listening for messages

serviceHost.Open();

Also in service configuration file you will need to add mapping

<host>

<baseAddresses>

<add baseAddress="http://localhost:9904/demoservice/" />

</baseAddresses>

</host>

Once these changes are done you should be able to run the application and you service would be available on the specified URL.

Please check attached sample project for reference that demonstrate creating self hosted WCF service

WCF,ASP.Net,Web Service,ASMX,WSDL