

WSD LAB-07

NAME : THUMBAR KEVIN BALUBHAI

ROLL NO. : CE145

Message Exchange Pattern : OneWay and Request-Reply :

WCFServiceMEP :

ServiceClassMEP.cs :

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.ServiceModel;
using System.Text;
using System.Threading;
using System.Threading.Tasks;

namespace WCFServiceMEP
{
    [ServiceBehavior(InstanceContextMode = InstanceContextMode.PerCall)]
    public class ServiceClassMEP : IServiceMEP
    {
        void IServiceMEP.OneWayMesssage()
        {
            Thread.Sleep(1000);
        }

        string IServiceMEP.RequestReplyPattern()
        {
            Thread.Sleep(1000);
            return "Current time of Service is " + DateTime.Now.ToString();
        }
    }
}
```

IServiceMEP.cs :

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.ServiceModel;
using System.Text;
using System.Threading.Tasks;

namespace WCFServiceMEP
{
    [ServiceContract]
    internal interface IServiceMEP
    {
        [OperationContract(IsOneWay = true)]
        void OneWayMesssage();

        [OperationContract(IsOneWay = false)]
        string RequestReplyPattern();
    }
}
```

WCF Host From :**Form1.cs :**

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.ServiceModel;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using WCFServiceMEP;
```

```

namespace WCF_Host_From_App
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void label1_Click(object sender, EventArgs e)
        {

        }

        ServiceHost sh = null;
        private void Form1_Load(object sender, EventArgs e)
        {
            sh = new ServiceHost(typeof(ServiceClassMEP));
            sh.Open();
            label1.Text = "server is running";
        }

    }
}

```

App.config :

```

<?xml version="1.0" encoding="utf-8"?>
<configuration>
    <startup>
        <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.8"/>
    </startup>
    <system.serviceModel>
        <services>
            <service name="WCFServiceMEP.ServiceClassMEP"
behaviorConfiguration="metadata">
                <host>
                    <baseAddresses>
                        <add
baseAddress="http://localhost:8733/Design_Time_addresses"/>

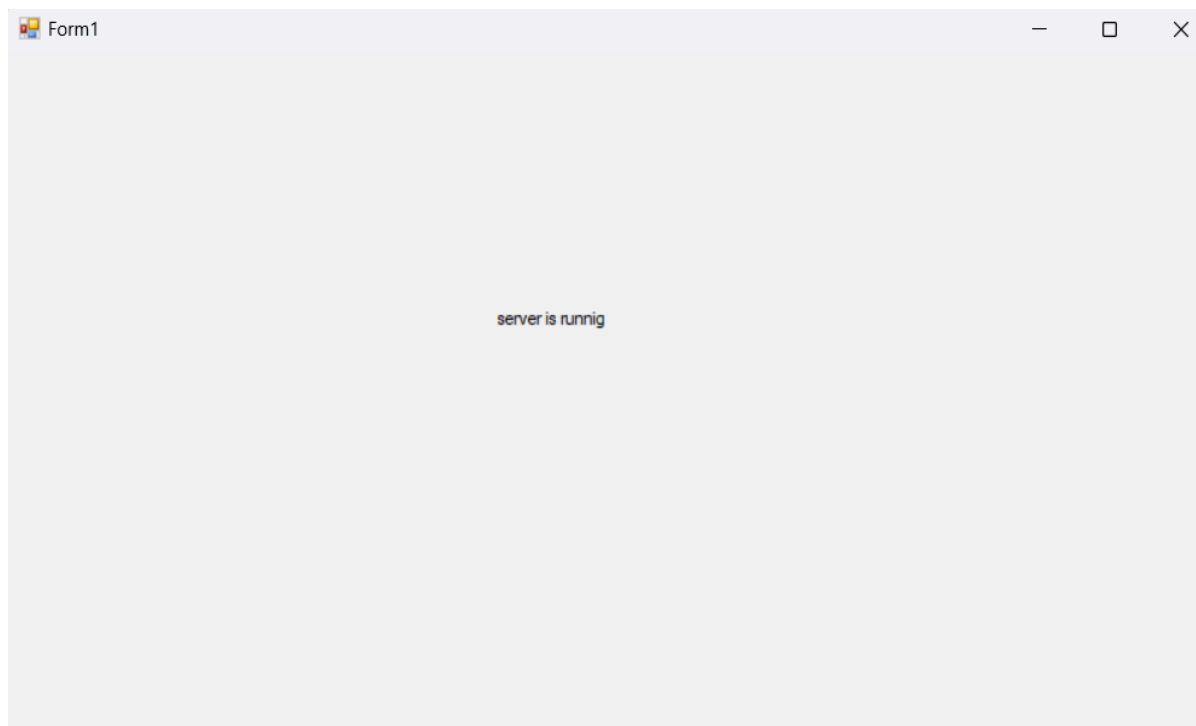
```

```

        </baseAddresses>
    </host>
    <endpoint address="" binding="basicHttpBinding"
contract="WCFServiceMEP.IServiceMEP"/>
    </service>
</services>
<behaviors>
    <serviceBehaviors>
        <behavior name="metadata">
            <serviceMetadata httpGetEnabled="True"
httpGetUrl=""/>
        </behavior>
    </serviceBehaviors>
</behaviors>
</system.serviceModel>
</configuration>

```

O/P :



WCF Client App :

Form1.cs :

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace WCF_Client_App
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void label1_Click(object sender, EventArgs e)
        {
        }

        private void button1_Click(object sender, EventArgs e)
        {
            label1.Text = "on the load time the time is " + DateTime.Now.ToString();
            ServiceReference1.ServiceMEPClient sc =
                new ServiceReference1.ServiceMEPClient("BasicHttpBinding_IServiceMEP");
            sc.OneWayMesssage();
            label2.Text = "After Calling the service the time is " + DateTime.Now.ToString();
        }

        private void button2_Click(object sender, EventArgs e)
        {
        }
    }
}
```

```

        label1.Text = "on the load time the time is " + DateTime.Now.ToString();
        ServiceReference1.ServiceMEPClient sc =
            new
ServiceReference1.ServiceMEPClient("BasicHttpBinding_IServiceMEP");
        sc.RequestReplyPattern();
        label2.Text = "After Calling the service the time is " + DateTime.Now.ToString();

    }
}
}

```

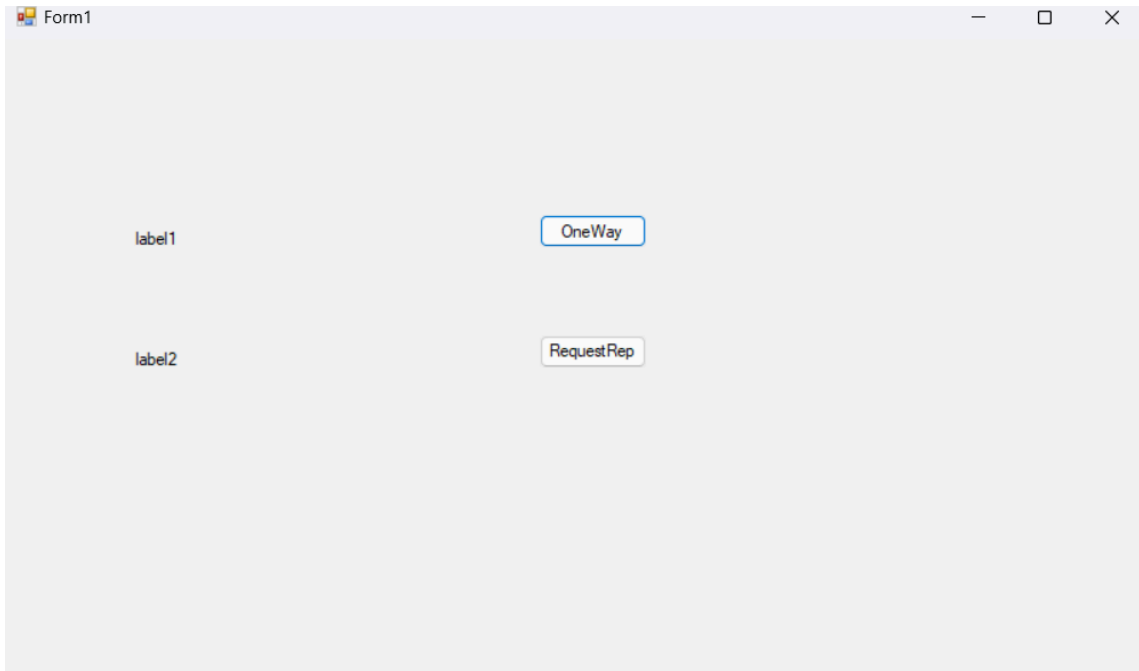
App.config :

```

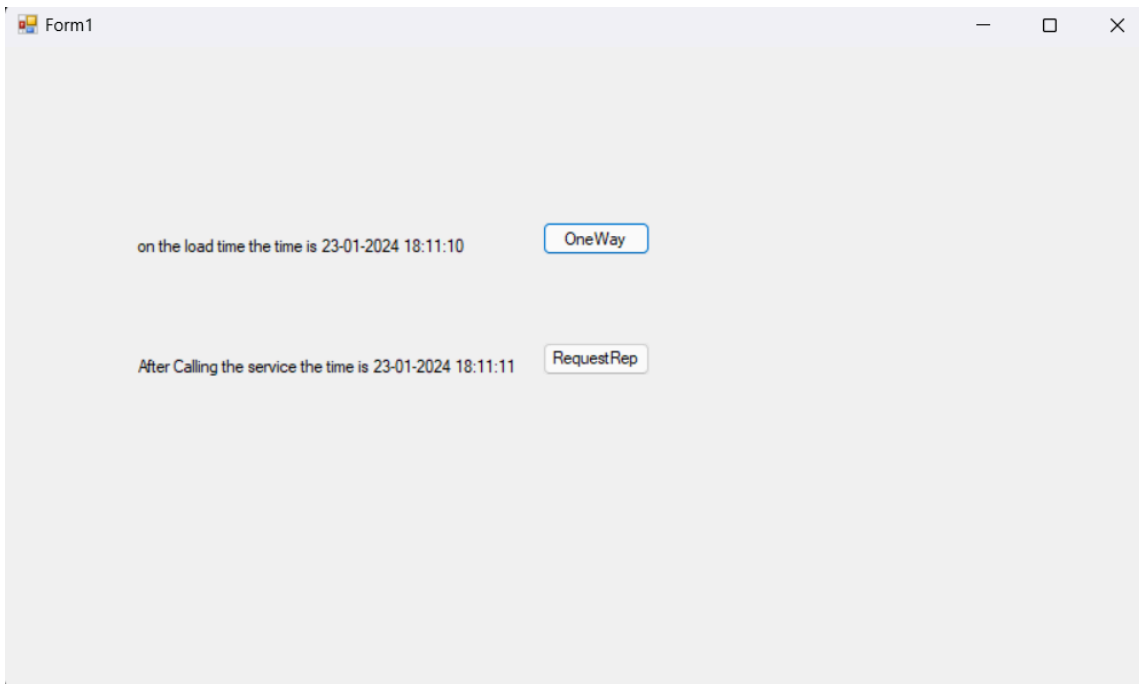
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.7.2" />
  </startup>
  <system.serviceModel>
    <bindings>
      <basicHttpBinding>
        <binding name="BasicHttpBinding_IServiceMEP" />
      </basicHttpBinding>
    </bindings>
    <client>
      <endpoint address="http://localhost:8733/Design_Time_addresses"
        binding="basicHttpBinding"
        bindingConfiguration="BasicHttpBinding_IServiceMEP"
        contract="ServiceReference1.IServiceMEP"
        name="BasicHttpBinding_IServiceMEP" />
    </client>
  </system.serviceModel>
</configuration>

```

O/P:



A screenshot of a Windows application window titled "Form1". The window contains two labels, "label1" and "label2", positioned on the left side. To the right of "label1" is a button labeled "OneWay". To the right of "label2" is a button labeled "RequestRep". The window has a standard Windows title bar with minimize, maximize, and close buttons.



A screenshot of a Windows application window titled "Form1". The window displays time information. The first line of text is "on the load time the time is 23-01-2024 18:11:10", followed by a button labeled "OneWay". The second line of text is "After Calling the service the time is 23-01-2024 18:11:11", followed by a button labeled "RequestRep". The window has a standard Windows title bar with minimize, maximize, and close buttons.

Form1

on the load time the time is 23-01-2024 18:11:19

OneWay

After Calling the service the time is 23-01-2024 18:11:20

RequestRep

Message Exchange Pattern : Duplex : -

WCFServiceMEP :

IServiceMEP.cs :

```
using System.ServiceModel;
```

```
namespace WCFServiceMEP_Duplex
```

```
{  
    [ServiceContract(CallbackContract = typeof(IServiceCallBack))]  
    public interface IServiceMEP  
    {  
        [OperationContract(IsOneWay = true)]  
        void hello();  
    }  
  
    public interface IServiceCallBack  
    {  
        void helloback(string name);  
    }  
}
```

ServiceClassMEP.cs :

```
using System.ServiceModel;
```

```
using System.Threading;
```

```
namespace WCFServiceMEP_Duplex
```

```
{  
    [ServiceBehavior(ConcurrencyMode = ConcurrencyMode.Reentrant)]  
    public class ServiceClassMEP : IServiceMEP  
    {  
        public void hello()  
        {  

```

```
Thread.Sleep(1000);  
string name = "Windows Communication Foundation";
```

```
OperationContext.Current.GetCallbackChannel<IServiceCallback>().helloback(n  
ame);  
    }  
}  
}
```

WCF Service Host :

Form1.cs :

```
using System;  
using System.Collections.Generic;  
using System.ComponentModel;  
using System.Data;  
using System.Drawing;  
using System.Linq;  
using System.ServiceModel;  
using System.Text;  
using System.Threading.Tasks;  
using System.Windows.Forms;  
  
namespace WCF_Host_From_App  
{  
    public partial class Form1 : Form  
    {  
        public Form1()  
        {  
            InitializeComponent();  
        }  
  
        private void label1_Click(object sender, EventArgs e)  
        {  

```

```

    }

    private void Form1_Load(object sender, EventArgs e)
    {
        ServiceHost sh = new
ServiceHost(typeof(WCFServiceMEP_Duplex.ServiceClassMEP));
        sh.Open();
        label1.Text = "Service Running !";
    }

}
}

```

App.config :

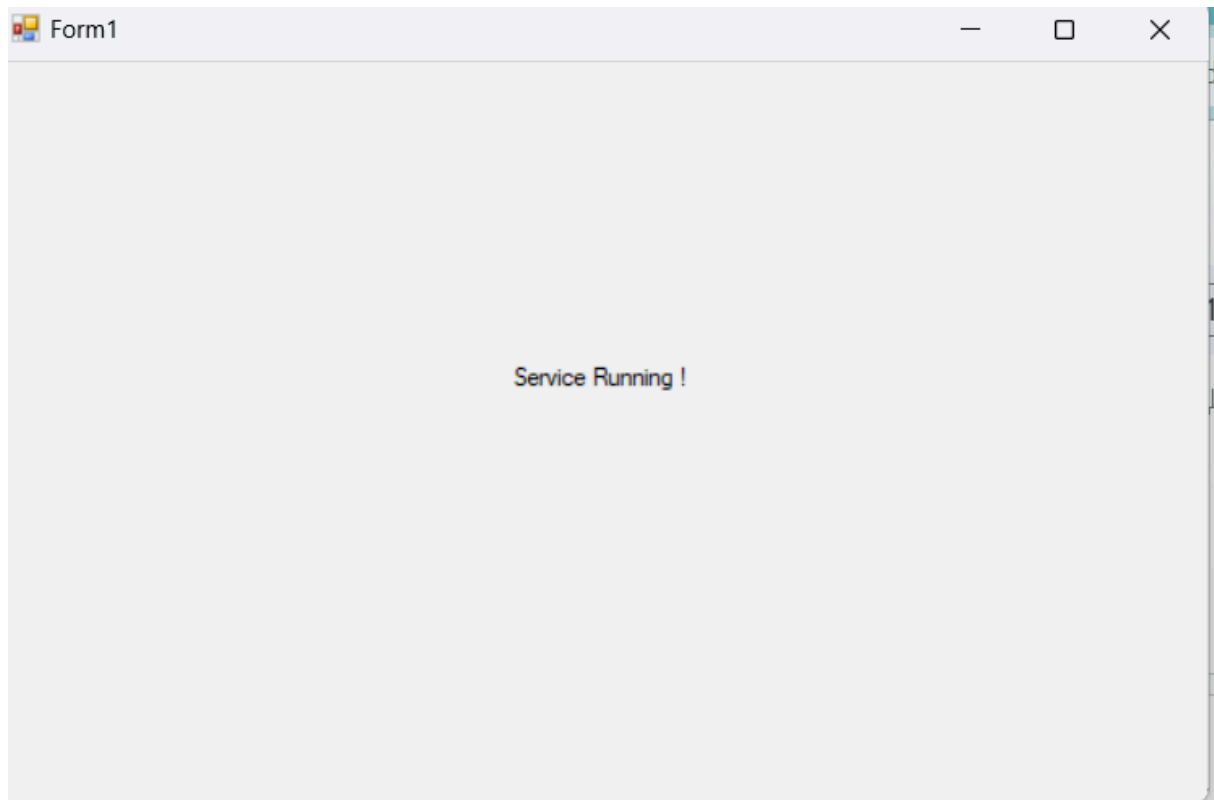
```

<?xml version="1.0" encoding="utf-8" ?>
<configuration>
    <system.serviceModel>
        <services>
            <service name="WCFServiceMEP_Duplex.ServiceClassMEP"
behaviorConfiguration="metadataSupport">
                <host>
                    <baseAddresses>
                        <add baseAddress =
"http://localhost:8733/Design_Time_Addresses" />
                    </baseAddresses>
                </host>
                <endpoint address="" binding="wsDualHttpBinding"
contract="WCFServiceMEP_Duplex.IServiceMEP"/>
            </service>
        </services>
        <behaviors>
            <serviceBehaviors>
                <behavior name ="metadataSupport">

```

```
        <serviceMetadata httpGetEnabled="True"/>
    </behavior>
</serviceBehaviors>
</behaviors>
</system.serviceModel>
</configuration>
```

O/P:



WCF Service Client :

Form1.cs :

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
```

```

using System.Data;
using System.Drawing;
using System.Linq;
using System.ServiceModel;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace WCF_Client_Form_App
{
    public partial class Form1 : Form, ServiceReference1.IServiceMEPCallBack
    {
        public Form1()
        {
            InitializeComponent();
        }

        public void helloback(string name)
        {
            label1.Text = name;
        }

        private void button1_Click(object sender, EventArgs e)
        {
            InstanceContext instance = new InstanceContext(this);
            ServiceReference1.ServiceMEPClient client = new
ServiceReference1.ServiceMEPClient(instance);
            client.hello();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
        }
    }
}

```

O/P :

