

Hosting WCF Service in Windows Server AppFabric

If you are developing distributed service oriented applications using Microsoft.NET, then the Windows Communication Foundation (WCF) framework is strongly recommended. WCF is a great choice for applications that communicate over various protocols e.g., HTTP, TCP, MSMQ etc. and also need a secure communication mechanism.

One of the major challenges a developer faces is how to monitor and track a WCF service, once the WCF service is hosted in the hosting environment. Once the application is deployed in the production environment, it becomes essential to track how the WCF service is performing e.g. while making a call to WCF if any exception occurs then how do we know about it and so on. To make such things possible, we have been provided a new hosting environment that is **Windows Server AppFabric**.

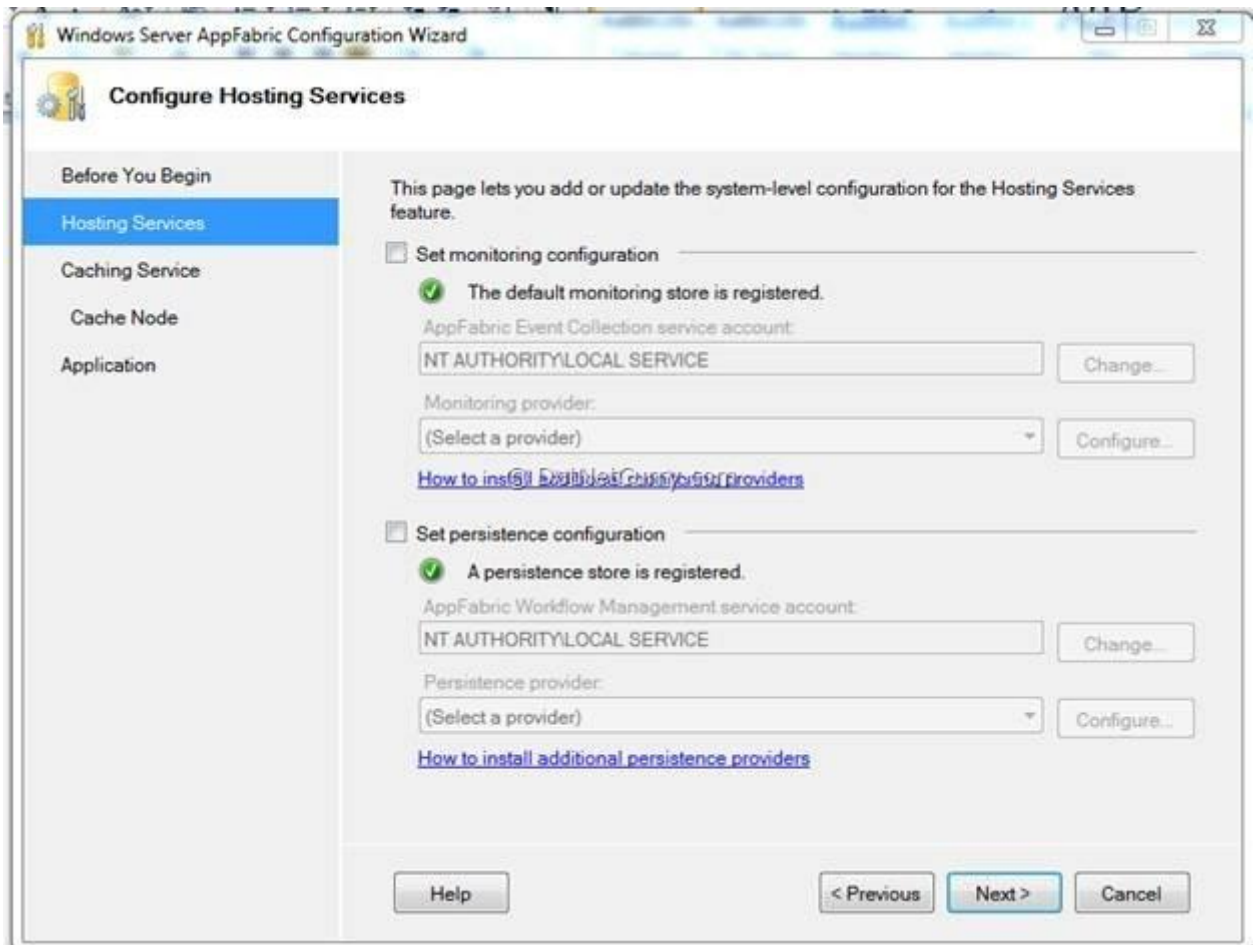
Windows Server AppFabric extends windows server to provide enhanced hosting, monitoring caching capabilities especially for WCF middle tier services and web applications. This is an extension to the IIS Web server (version 7) and requires .NET Framework 4.0. It provides features like Hosting Services, tools to easy deploy, configure and monitor WCF services and Workflow Services.

Windows Server AppFabric provides two runtime databases - Monitoring Schema and Persistence Schema. Monitoring Schema provides capabilities to monitor activities of the WCF and WF services hosted. Persistence Schema provides capabilities to save state of the Workflow running. After installing the AppFabric, you need to configure it for Monitoring and Persistence and while doing this, the required database gets created. Here you need to make sure that you meet the following system requirements:

- Windows Server 2008 32 or 64 Bit. Or Windows 7 32 or 64 bit.
- IIS 7.x
- Sql Server 2008 (R2 is better).
- MS.NET 4.0

While installing the AppFabric, you need to have Admin rights. You can download [Windows Server AppFabric from here](#) (Please choose the correct installable.)

After installing AppFabric, you need to run the configuration wizard for monitoring and persistence as below.



Create a Web Site in IIS 7 targeted to .NET 4.0. You can find similar steps for creating Web Site on IIS 7 using the link below:

[Hosting WCF 4.0 Service on IIS 7.5 with SSL](#) (Just ignore the SSL settings, rest remains the same)

Creating WCF Service to be Hosted in IIS 7 and Monitoring using AppFabric

Step 1: Open VS2010 and create a WCF Service Application. Rename IService1.cs to IService.cs and Service1.Svc to Service.svc. Open IService.cs and add the below ServiceContract and DataContract.

```

using System.Runtime.Serialization;
using System.ServiceModel;

namespace WCF_DbService
{
    [ServiceContract]
    public interface IService
    {
        [OperationContract]
        Department[] GetDepartments(string dbName);
    }

    © DotNetCurry.com

    [DataContract]
    public class Department
    {
        [DataMember]
        public int DeptNo { get; set; }
        [DataMember]
        public string DeotName { get; set; }
        [DataMember]
        public string Location { get; set; }
    }
}

```

Step 2: Open Service.svc.cs and add the following code in it:

```

using System.Data;
using System.Data.SqlClient;
namespace WCF_DbService
{
    public class Service : IService
    {
        SqlConnection Conn;
        SqlCommand Cmd;
        SqlDataReader Reader;

        public Department[] GetDepartments(string dbName)
        {
            Conn = new SqlConnection("Data Source=.;Initial
            Catalog="+dbName+";Integrated Security=SSPI");
            Conn.Open();
            Cmd = new SqlCommand("Select * from Department", Conn);
            Reader = Cmd.ExecuteReader();
            DataTable DtDept = new DataTable();
            if (Reader.HasRows)
            {
                DtDept.Load(Reader);
            }
            Department[] Departments = new Department[DtDept.Rows.Count];
            int i = 0;
            foreach (DataRow Dr in DtDept.Rows)
            {
                Departments[i] = new Department();
                Departments[i].DeptNo =
                    Convert.ToInt32(Dr["DeptNo"]);
                Departments[i].DeptName =
                    Dr["Dname"].ToString();
                Departments[i].Location =
                    Dr["Location"].ToString();
                i = i + 1;
            }
            return Departments;
        }
    }
}

```

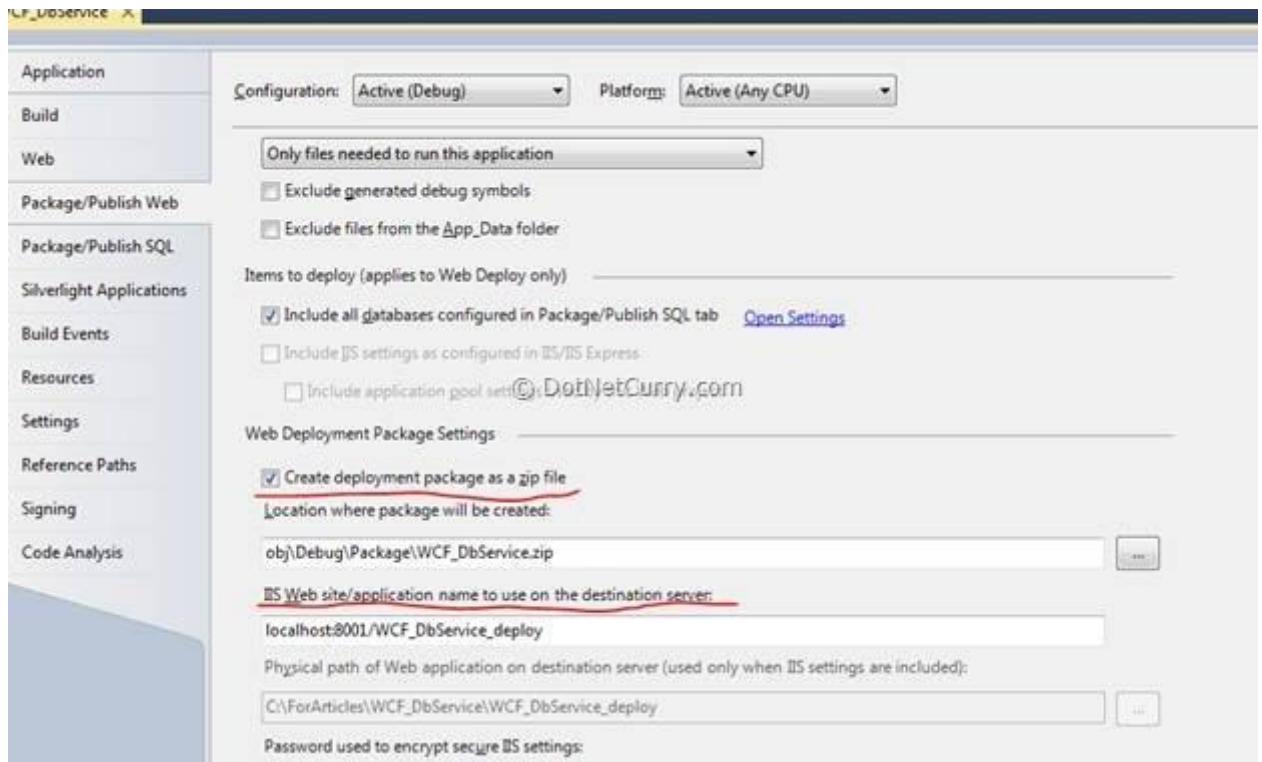
Step 3: Change the Service.svc in Markup as below:

```

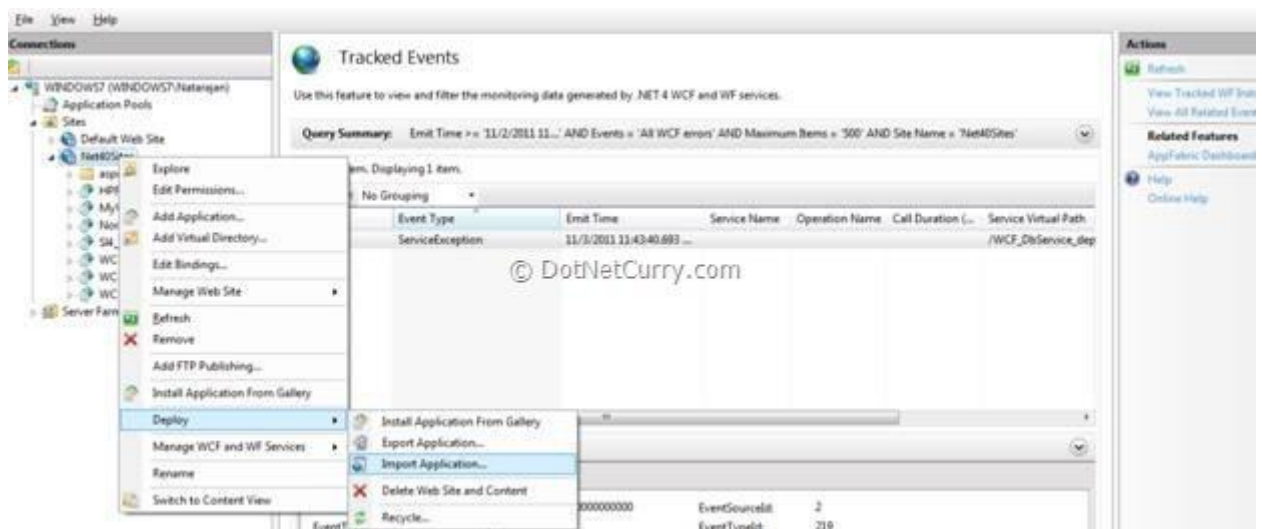
<%@ ServiceHost Language="C#" Debug="true" Service="WCF_DbService.Service"
CodeBehind="Service.svc.cs" %>

```

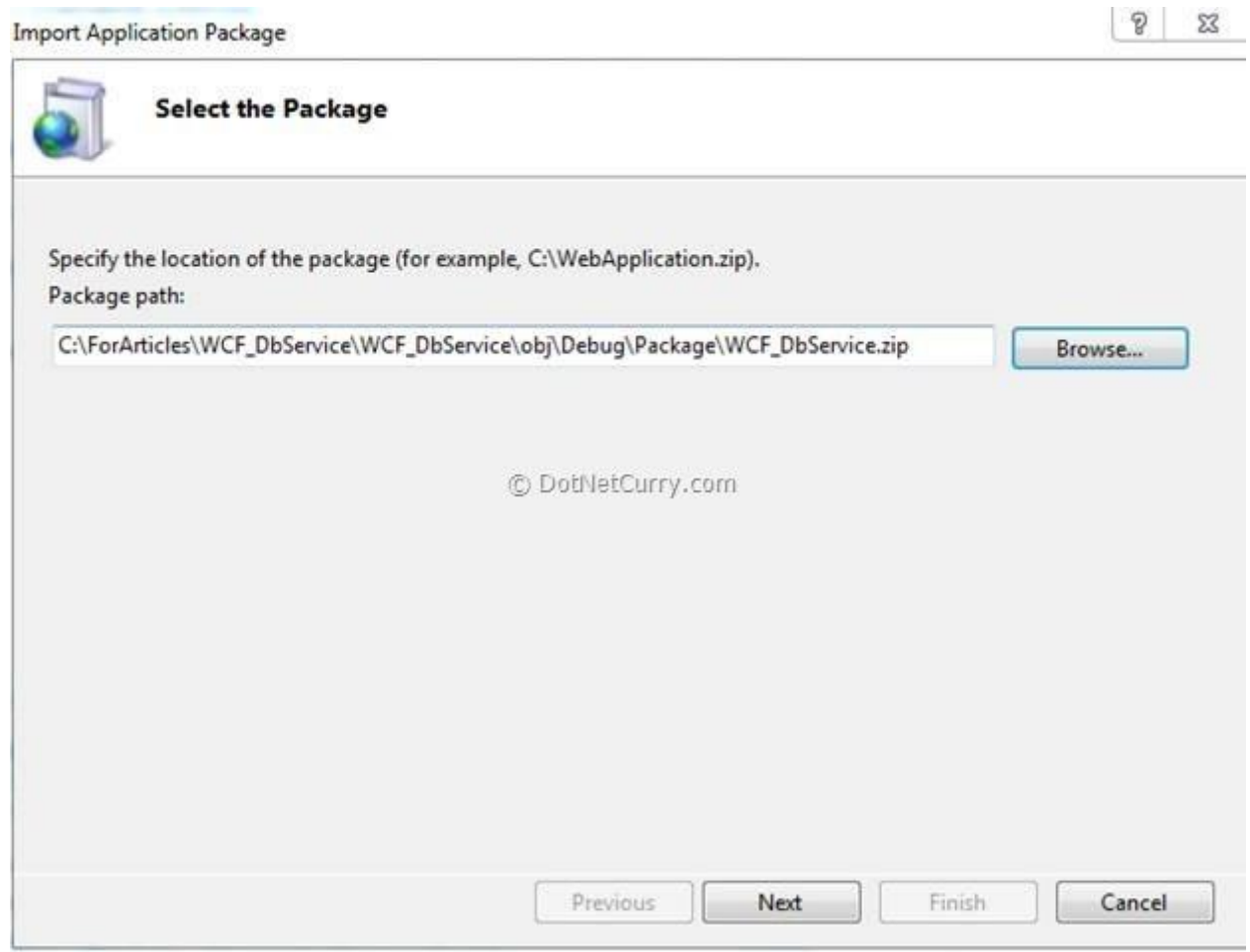
Step 4: Right-Click on the WCF Application and go to properties, Select Package/Publish Web, make sure that 'Create a Deployment package as a Zip file'. This will specify the location of the Zip file where the deployment package is stored. Also specify the Web Site name on the destination host server as below:



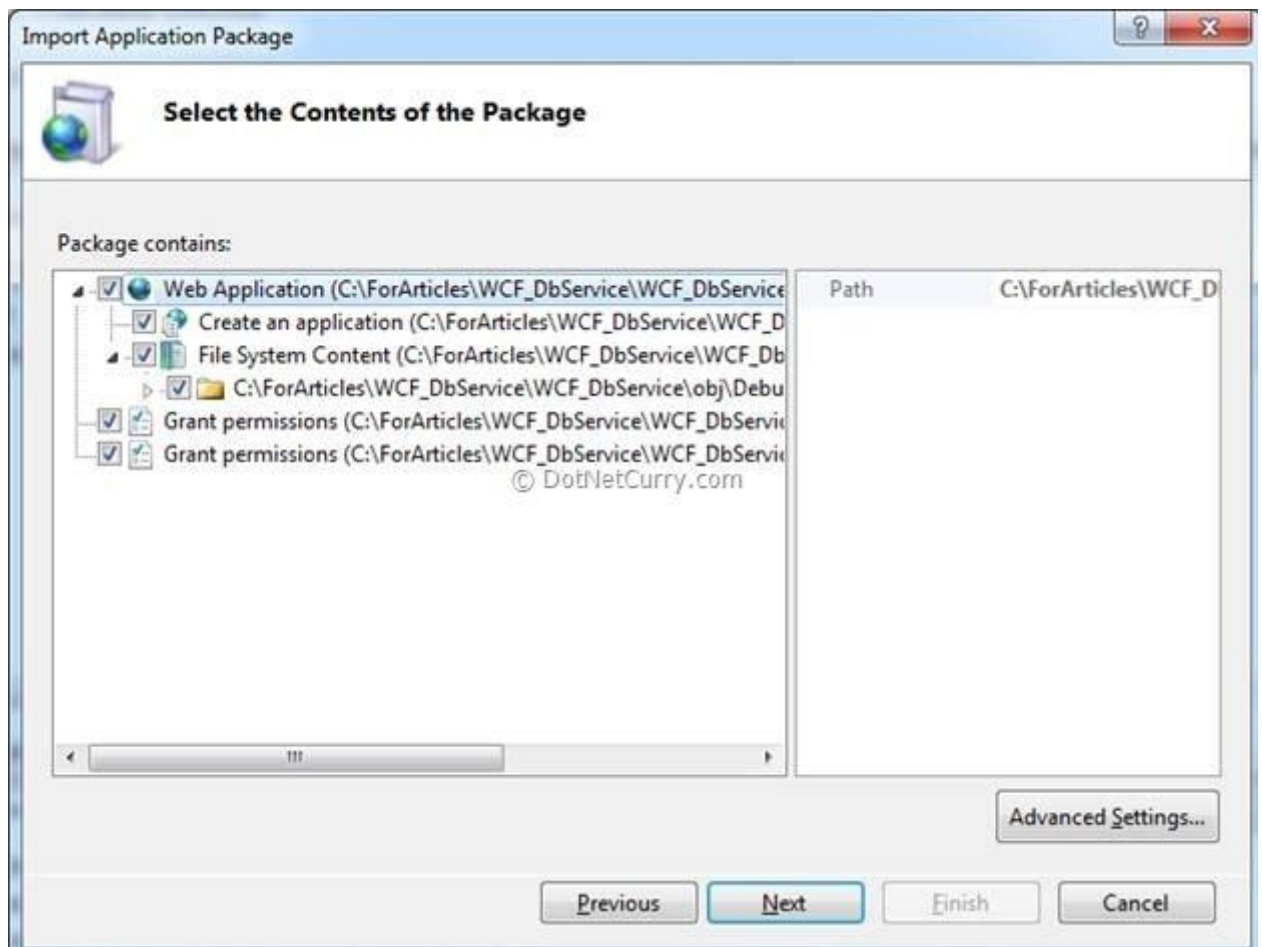
Step 5: Open IIS, right click on the Web Site and Select Deploy > Import Application as shown below:



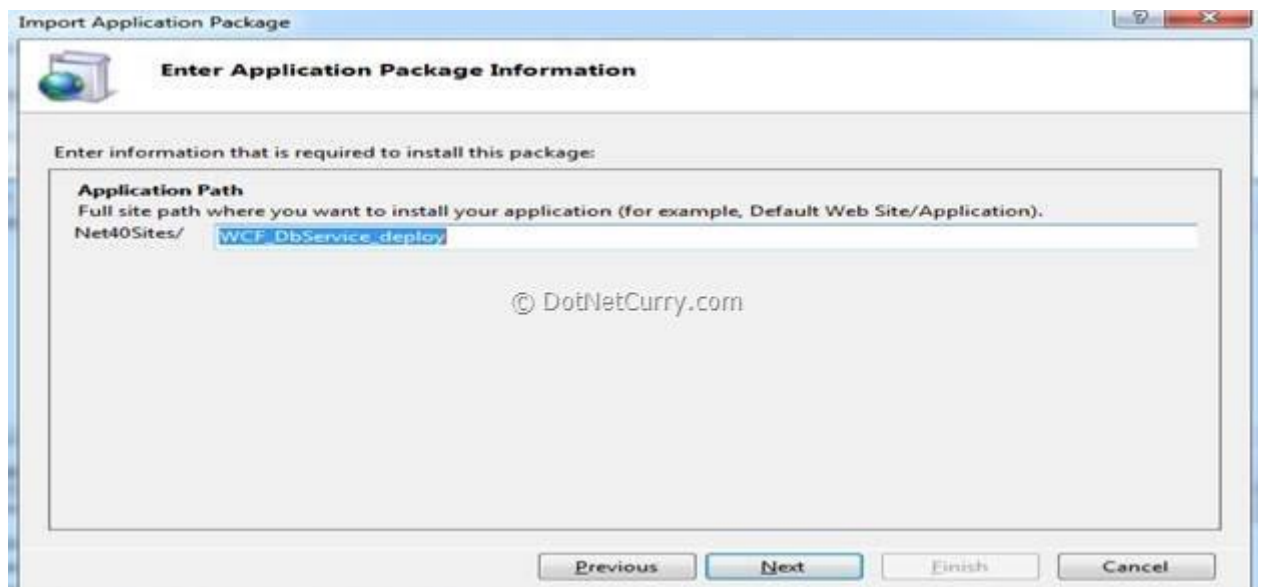
Step 6: Select the package path as below and click Next:



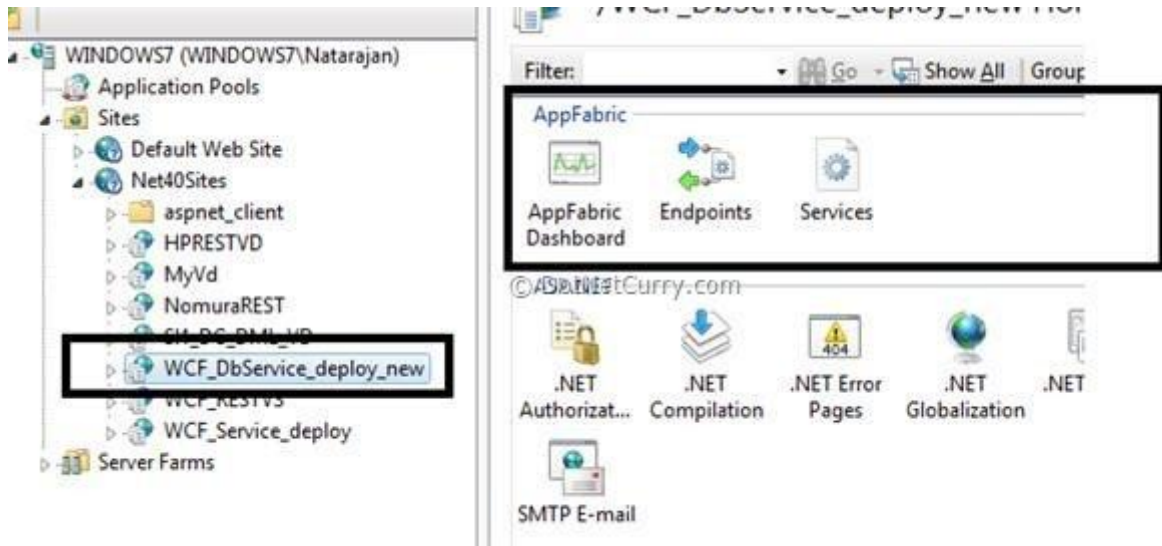
Step 7: The Next step shows the Package contents, keep the selection as it is and click Next:



The application path is as below:



The web application gets created in IIS:



The marked area represents the Web Application created by the name of 'WCF_DbService_deploy_new' and the AppFabric Dashboard.

Step 8: Open the WCF Service in the browser by browsing the Service.Svc file from the 'Content View'. The result will be as below:

Service Service

You have created a service.

To test this service, you will need to create a client and use it to call the service. You can do this using the svcutil.exe tool from the command line with the following:

```
svcutil.exe http://localhost:8001/WCF_DbService_deploy_new/Service.svc?wsdl
```

This will generate a configuration file and a code file that contains the client class. Add the two files to your client application and use the generated client class to

C#

```
class Test
{
    static void Main()
    {
        ServiceClient client = new ServiceClient();

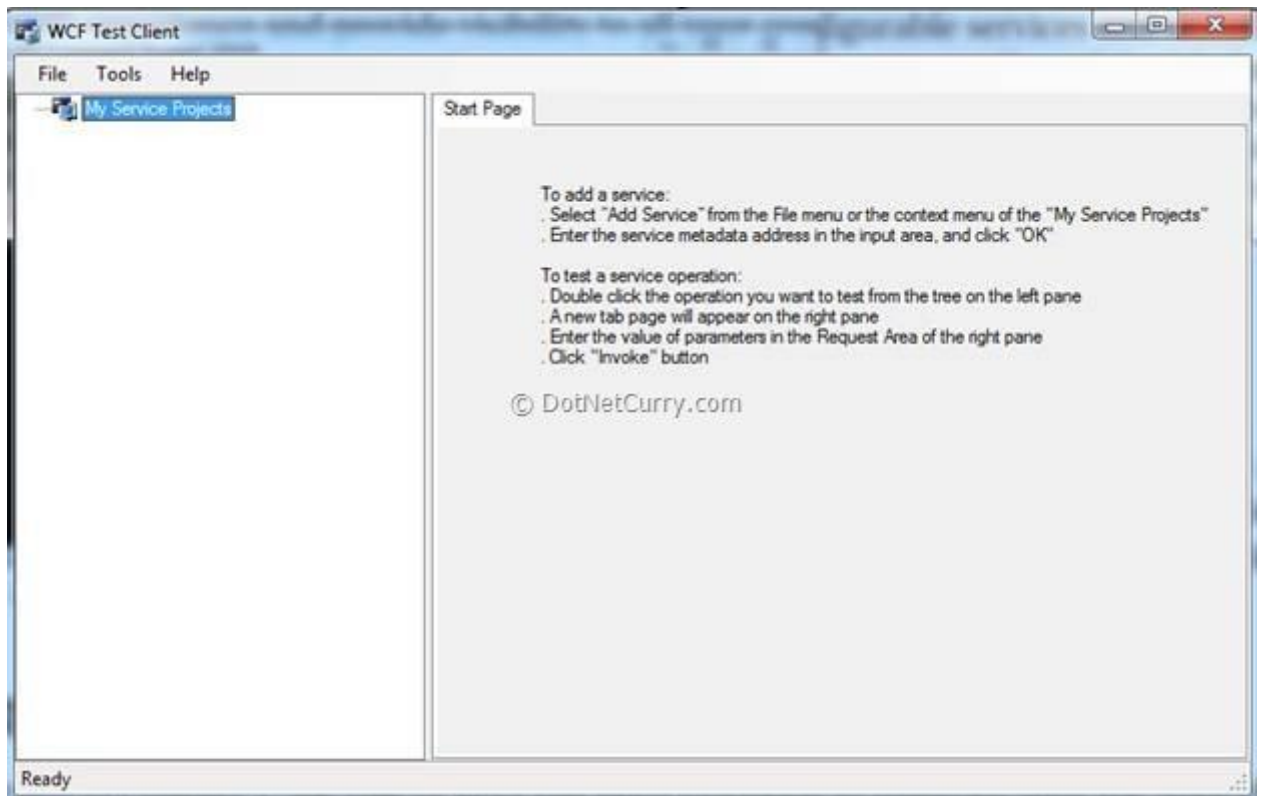
        // Use the 'client' variable to call operations on the service.
        // Always close the client.
        client.Close();
    }
}
```

Visual Basic

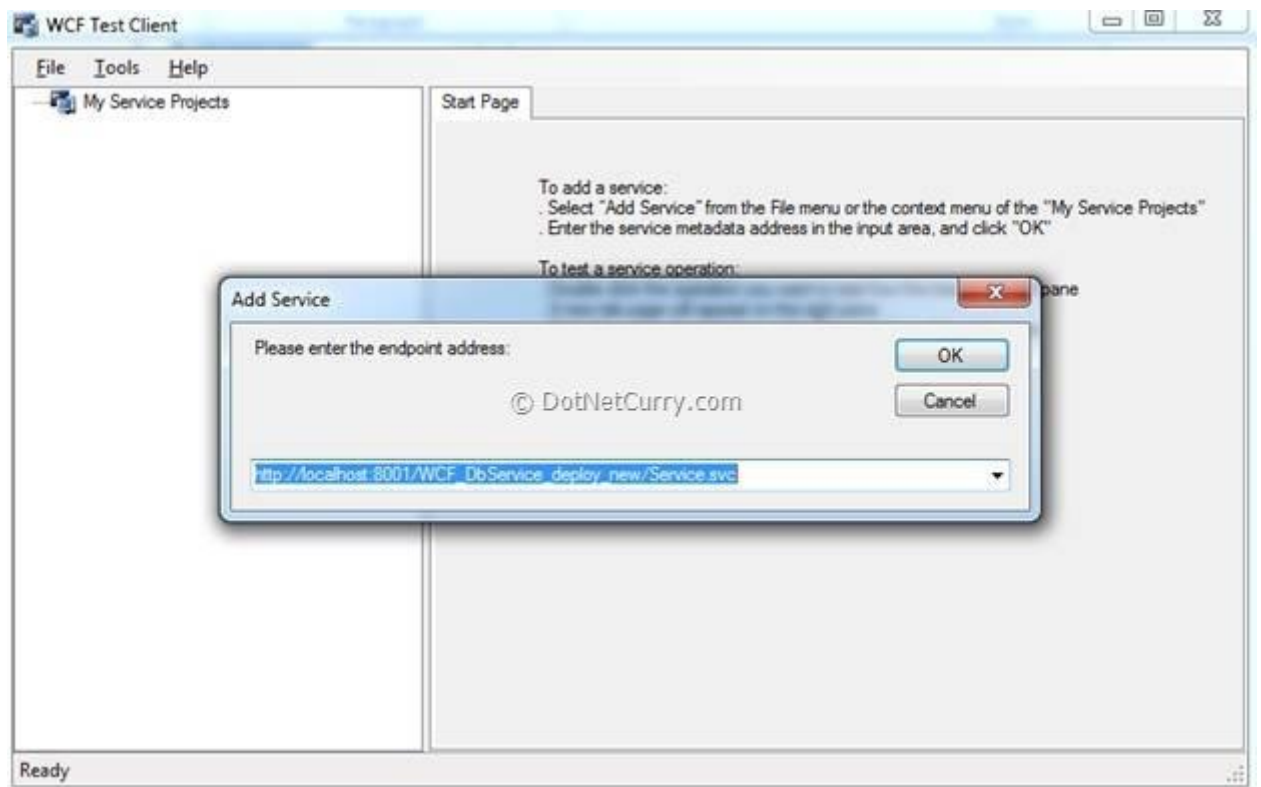
```
Class Test
    Shared Sub Main()
        Dim client As ServiceClient = New ServiceClient()
        ' Use the 'client' variable to call operations on the service.

        ' Always close the client.
        client.Close()
    End Sub
End Class
```

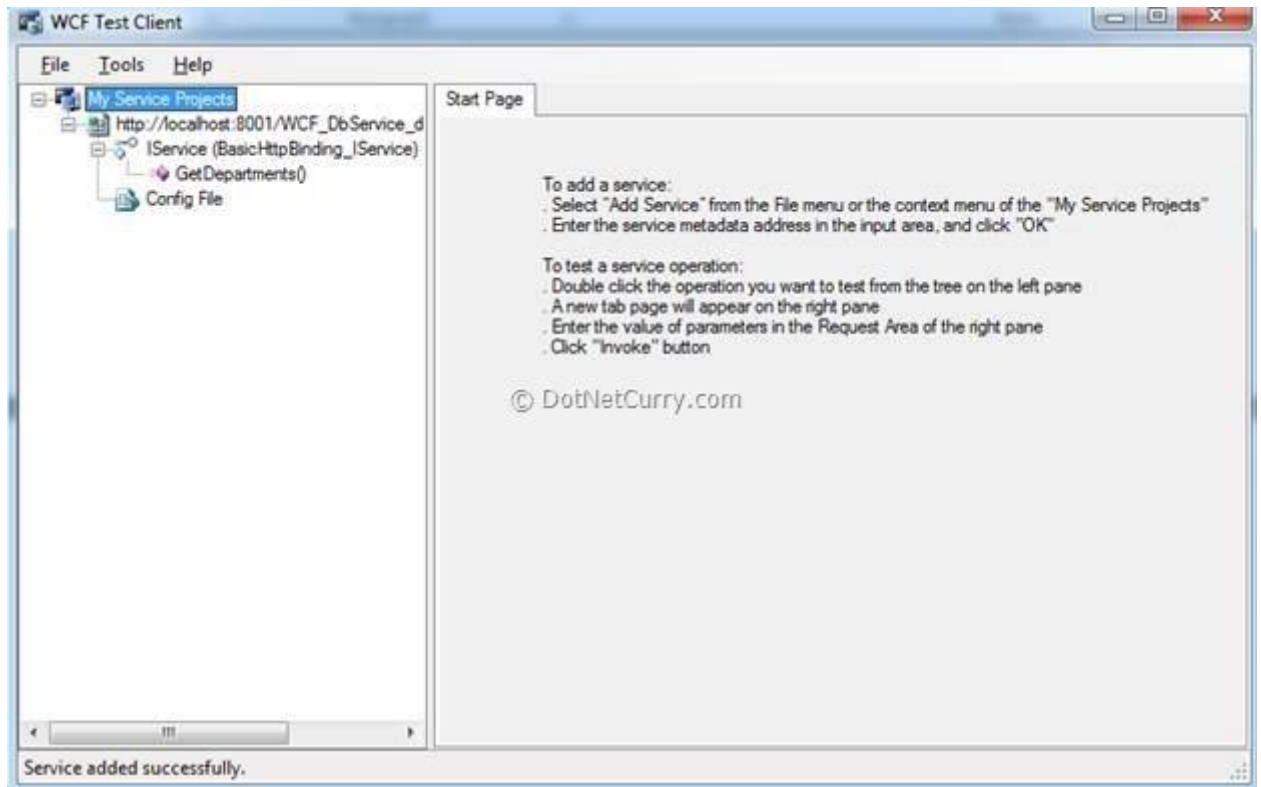
Step 9: Go to the .NET Command Prompt and type the following in the Command-line - 'WCFTestClient'. This will provide the default Client Testing as shown below:



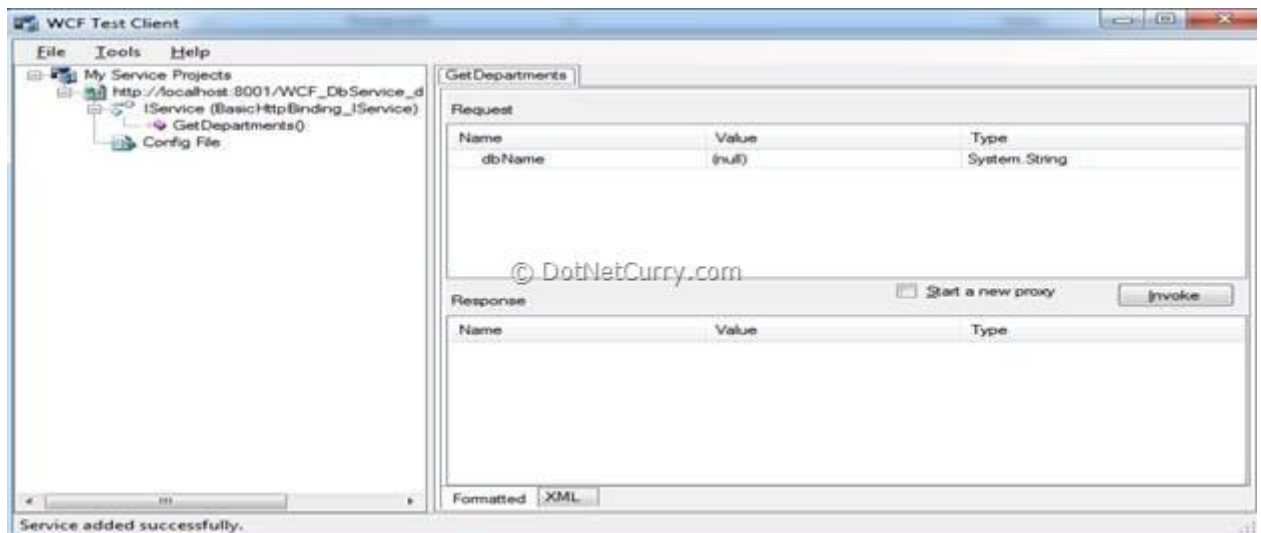
Step 10: Right click on the 'My Service Projects', and type the EndPoint address as shown below:



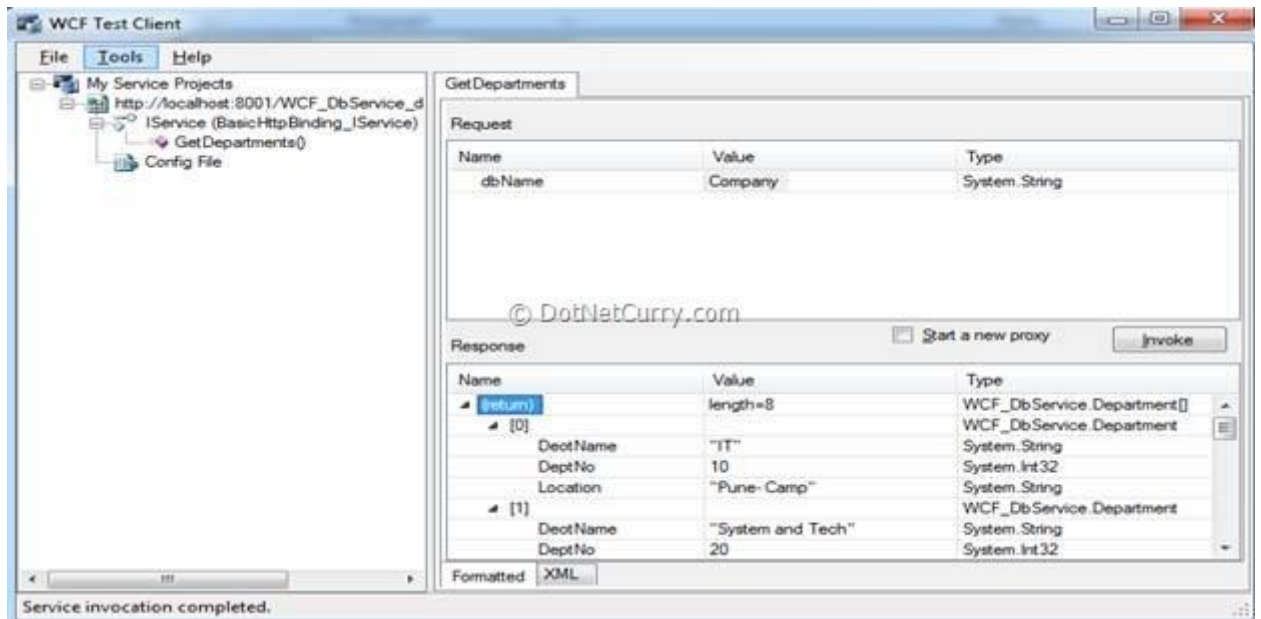
You will get the service endpoint information as shown below:



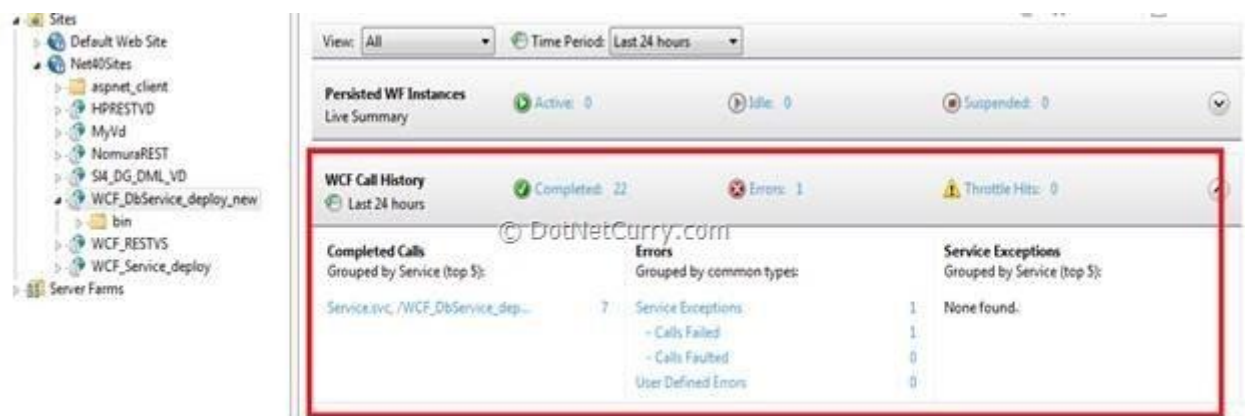
Step 11: Double Click the 'GetDepartments()' method and you will get the display as shown below:



and enter the name of the Database e.g. 'Company' and click on invoke, the result will be displayed in the Response part as shown below:



Step 12: Since, the WCF service is hosted in IIS and AppFabric is configured for IIS, select the application you have created and then click on the 'AppFabric DashBoard'. The result will be as below:



The RED marked area represents the WCF service monitoring statistics i.e Completed calls and Errors, if any. If you click on 'Completed', the Track Events windows gets displayed. This window provides the summary of the calls as below:

Tracked Events

Use this feature to view and filter the monitoring data generated by .NET 4 WCF and WF services.

Query Summary: Emit Time >= '11/2/2011 1:...' AND Events = 'WCF complete...' AND Maximum Items = '500' AND Site Name = 'Net40Sites' AN...

Found 11 items. Displaying 11 items.

Group by: Machine

Level	Event Type	Emit Time	Service Name	Operation Name	Call Duration (...)	Service Virtual Pat
Information	AggregateOperationComple...	11/3/2011 11:42:21.301 ...	Service	GetDepartments	929	/WCF_DbService_...
Information	AggregateOperationComple...	11/3/2011 11:43:21.395 ...	Service	Get	0	/WCF_DbService_...
Information	AggregateOperationComple...	11/3/2011 11:42:11.281 ...	Service	Get	2	/WCF_DbService_...
Information	AggregateOperationComple...	11/3/2011 11:41:41.217 ...	Service	Get	1406	/WCF_DbService_...
Information	AggregateOperationComple...	11/3/2011 11:40:11.217 ...	Service	Get	0	/WCF_DbService_...
Information	AggregateOperationComple...	11/3/2011 11:43:21.395 ...	Service	Get	0	/WCF_DbService_...
Information	AggregateOperationComple...	11/3/2011 1:25:02.762 PM	Service	Get	0	/WCF_DbService_...
Information	AggregateOperationComple...	11/3/2011 1:27:53.127 PM	Service	GetDepartments	1086	/WCF_DbService_...

Details:

Overview | Tracked Variables (0) | Errors (0)

AggregateCount:	1	AverageDuration:	929
EventSourceId:	1	EventType:	AggregateOperationCompleted
EventTypeId:	364	MaxDuration:	929
OperationName:	GetDepartments	ProcessId:	4500
TimeCreated:	11/3/2011 11:42:21.301 AM	TraceLevel:	Information

While testing the call using the 'WCFTestClient', if you enter any invalid value, an exception will be occur and in the AppFabric, the error summary will be displayed, as below:

Tracked Events

Use this feature to view and filter the monitoring data generated by .NET 4 WCF and WF services.

Query Summary: Emit Time >= '11/2/2011 1:...' AND Events = 'All WCF errors' AND Maximum Items = '500' AND Site Name = 'Net40Sites' AND P...

Found 1 item. Displaying 1 item.

Group by: Machine

Level	Event Type	Emit Time	Service Name	Operation Name	Call Duration (...)	Service Virtual Path
Error	ServiceException	11/3/2011 11:43:40.693 ...				/WCF_DbService_dep

Details:

Overview | Tracked Variables (0) | Errors (1)

E2EActivityId:	00000000-0000-0000-0000-000000000000	EventSourceId:	2
EventType:	ServiceException	EventTypeId:	219
ExceptionTypeName:	System.Data.SqlClient.SqlException	ProcessId:	4500
TimeCreated:	11/3/2011 11:43:40.693 AM	TraceLevel:	Error

Conclusion: The Windows Server AppFabric is a must have for an administrator to monitor WCF and WF Services, for effective application management.