**WCF SERVICE PRIMER**

Version: 1.0

Khanh Nguyen

1. **Prerequisite:**

WCF Service is created with an interface. A sample WCF Service Interface would look something like this:

[ServiceContract]

public interface IApplicantService

{

[OperationContract]

Applicant GetApplicantByID(int id);

[OperationContract]

Applicant GetApplicantByAppliedID(int id);

[OperationContract]

Applicant GetApplicants();

}

A Service Contract specifies a service that will be implemented later on in a separate file. The Operation Contract exposes methods that a service client can use. To implement this service, you would do something like this:

public class AESApplicationService : IApplicationService, IApplicantService

{

public Applicant GetApplicantByID(int applicantID)

{

throw new NotImplementedException();

// TODO: This is where you would put your implementation

}

public Applicant GetApplicantByAppliedID(int id)

{

throw new NotImplementedException();

// TODO: This is where you would put your implementation

}

public Applicant GetApplicants()

{

throw new NotImplementedException();

// TODO: This is where you would put your implementation

}

}

However, there’s one problem with our service implementation. The service client wouldn’t know Applicant object. In order for the client to be able to recognize the object. We have to define a Data Contract attribute for the Applicant object. For each of the properties of that object, we have to specify Data Member attribute. These properties then will be exposed to the client.

[DataContract]

public partial class Applicant

{

[DataMember]

public int Applicant\_ID { get; set; }

[DataMember]

public string FirstName { get; set; }

[DataMember]

public string LastName { get; set; }

[DataMember]

public string SSN { get; set; }

[DataMember]

public string Gender { get; set; }

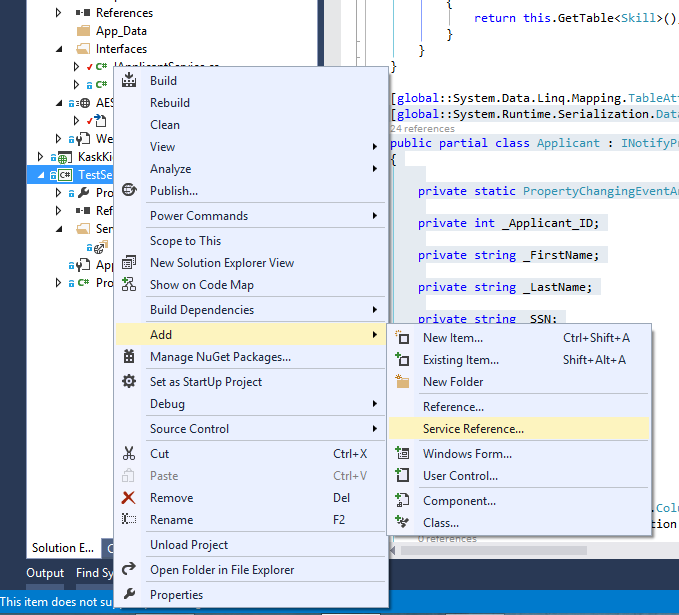
}

In our project, this is already defined since we are using LINQ to SQL and these objects are automatically marked as Data Contract. So when we added a DLL reference to the service as well as the client, we will be able to use Applicant object as well as other objects of the database.

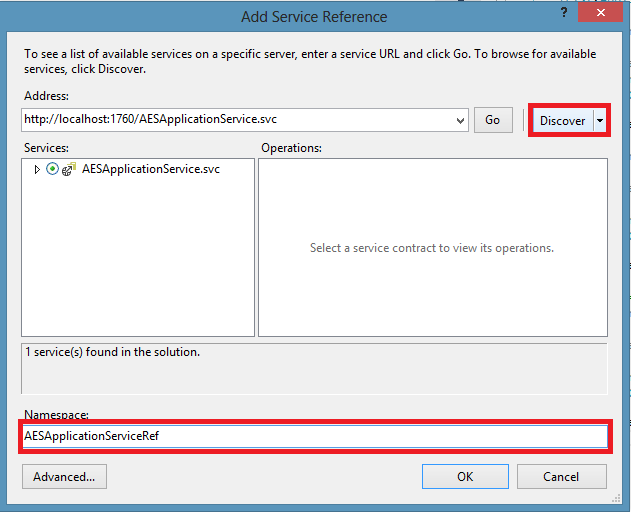
1. **Service Client:**

These are the steps you would do to make a service client in another project.

Create a new project (like usual). Right click on the newly created project and add a service reference:



Now on the next screen, the first step is to click on Discover then specify a name for your client:



1. **Consuming WCF Service:**

Now in your new project, when you have your service reference. You would do this to use it:

using TestService.AESServiceRef;

namespace TestService

{

class Program

{

static void Main(string[] args)

{

ApplicationServiceClient client = new ApplicationServiceClient();

ApplicantServiceClient client2 = new ApplicantServiceClient();

IList<Application> apps = client.GetApplications();

Console.ReadLine();

}

}

}

Notice that you would have two service clients since your AESApplicationService implements 2 interfaces (which are your Service Contracts.) Each of these will have their own methods to use (these are defined in your interface.)