

Server side service.cs code:

Code Snippet

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.Serialization;
using System.ServiceModel;
using System.Text;
using System.Data;
using System.Data.Common;
using System.Data.SqlClient;
using Microsoft.Synchronization;
using Microsoft.Synchronization.Data;
using Microsoft.Synchronization.Data.Server;
using System.Collections;
using System.Collections.ObjectModel;
using System.IO;
using System.Collections.Specialized;
using System.Runtime.Serialization.Formatters.Binary;

// NOTE: If you change the class name "Service" here, you must also update
the reference to "Service" in Web.config and in the associated .svc file.
public class Service : IService
{
    static DbServerSyncProvider serverSyncProvider;
    public Service()
    {
        if (serverSyncProvider == null)
        {
            CreateServerProvider();
        }
    }
    private void CreateServerProvider()
    {
        // assume default sql server instance
        string sqlserver = Environment.MachineName;
        string workingDb = "test";
        string ocsTable = "table1";
        SqlConnection conn;
        SqlSyncAdapterBuilder builder;
        conn = new SqlConnection(string.Format("server = {0}; database = {1};
integrated security = true", sqlserver, workingDb));
        conn.Open();
        serverSyncProvider = new DbServerSyncProvider();
        serverSyncProvider.Connection = conn;
        builder = new SqlSyncAdapterBuilder();
        builder.Connection = conn;
        builder.SyncDirection = SyncDirection.Bidirectional;
        builder.TableName = ocsTable;
        builder.CreationTrackingColumn = "CreationDate";
        builder.UpdateTrackingColumn = "LastEditDate";
    }
}
```

```

builder.DeletionTrackingColumn = "DeletionDate";
builder.TombstoneTableName = ocsTable + "_Tombstone";
serverSyncProvider.SyncAdapters.Add(builder.ToSyncAdapter());
SqlCommand selectNewAnchorCmd = new SqlCommand();
selectNewAnchorCmd.CommandText = "SELECT @" +
SyncSession.SyncNewReceivedAnchor + " = GETUTCDATE()";
SqlParameter newRecAnchor = new SqlParameter();
newRecAnchor.ParameterName = "@" + SyncSession.SyncNewReceivedAnchor;
newRecAnchor.DbType = System.Data.DbType.DateTime;
newRecAnchor.Direction = System.Data.ParameterDirection.Output;
selectNewAnchorCmd.Parameters.Add(newRecAnchor);
serverSyncProvider.SelectNewAnchorCommand = selectNewAnchorCmd;
}
public string GetData(int value)
{
return string.Format("You entered: {0}", value);
}

public CompositeType GetDataUsingDataContract(CompositeType composite)
{
if (composite.BoolValue)
{
composite.StringValue += "Suffix";
}
return composite;
}
public SyncServerInfo GetServerInfo(SyncSession syncSession)
{
try
{
Logger.Log("In GetServerInfo()");
return serverSyncProvider.GetServerInfo(syncSession);
}
catch (Exception e)
{
Logger.Log(e.ToString());
throw e;
}
}
public SyncContext
ApplyChanges(Microsoft.Synchronization.Data.SyncGroupMetadata groupMetadata,
DataSet dataSet, SyncSession syncSession)
{
try
{
Logger.Log("In ApplyChanges()");
return serverSyncProvider.ApplyChanges(groupMetadata, dataSet, syncSession);
}
catch (Exception e)
{
Logger.Log(e.ToString());
}
}

```

```

throw e;
}
}
public SyncContext GetChanges(SyncGroupMetadata groupMetadata, SyncSession
syncSession)
{
SyncContext context = null;
try
{
Logger.Log("In EnumerateChanges()");
context = serverSyncProvider.GetChanges(groupMetadata, syncSession);
}
catch (Exception e)
{
Logger.Log(e.ToString());
throw e;
}
return context;
}
public SyncSchema GetSchema(string [] tableNames, SyncSession syncSession)
{
try
{
Logger.Log("In GetSchema()");
Collection<string> tables = new Collection<string>();
for (int i = 0; i < tableNames.Length; i++)
tables.Add(tableNames[i]);
return serverSyncProvider.GetSchema(tables, syncSession);
}
catch (Exception e)
{
Logger.Log(e.ToString());
ApplicationException newex = new ApplicationException("failed in
GetSchema()", e);
throw newex;
}
}
}
class Logger
{
static Object thisLock = new Object();
static string logfile = @"\\wcfctestlog.log";
public static void Log(string msg)
{
lock (thisLock)
{
StreamWriter sw = File.AppendText(logfile);
sw.WriteLine(DateTime.Now.ToString() + " : " + msg);
sw.Flush();
sw.Dispose();
}
}
}

```

```
}  
}
```

Server side IService.cs code:

Code Snippet

```
using System;  
using System.Linq;  
using System.Runtime.Serialization;  
using System.ServiceModel;  
using System.Text;  
using System.Data;  
using System.Data.Common;  
using System.Data.SqlClient;  
using Microsoft.Synchronization;  
using Microsoft.Synchronization.Data;  
using Microsoft.Synchronization.Data.Server;  
using System.Collections;  
using System.Collections.Generic;  
using System.Collections.ObjectModel;  
// NOTE: If you change the interface name "IService" here, you must also  
// update the reference to "IService" in Web.config.  
[XmlSerializerFormat]  
[ServiceContract]  
public interface IService  
{  
    [OperationContract]  
    string GetData(int value);  
    [OperationContract]  
    CompositeType GetDataUsingDataContract(CompositeType composite);  
    // TODO: Add your service operations here  
    #region sync service Server provider abstract methods  
    [OperationContract]  
    SyncServerInfo GetServerInfo(SyncSession syncSession);  
    [OperationContract]  
    SyncContext ApplyChanges(Microsoft.Synchronization.Data.SyncGroupMetadata  
groupMetadata, DataSet dataSet, SyncSession syncSession);  
    [OperationContract]  
    SyncContext GetChanges(SyncGroupMetadata groupMetadata, SyncSession  
syncSession);  
    [OperationContract]  
    SyncSchema GetSchema(string [] tableNames, SyncSession syncSession);  
    #endregion  
}  
// Use a data contract as illustrated in the sample below to add composite  
// types to service operations.  
[DataContract]  
public class CompositeType  
{  
    bool boolValue = true;
```

```

string stringValue = "Hello ";
[DataMember]
public bool BoolValue
{
    get { return boolValue; }
    set { boolValue = value; }
}
[DataMember]
public string StringValue
{
    get { return stringValue; }
    set { stringValue = value; }
}
}

```

in the web.config file for your WCF service: make sure the basicHttpBinding is used.

Code Snippet

```

<system.serviceModel>
<services>
<service name="Service" behaviorConfiguration="ServiceBehavior">
<endpoint address="" binding="basicHttpBinding" contract="IService">

<behaviors>
<serviceBehaviors>
<behavior name="ServiceBehavior">
<serviceMetadata httpGetEnabled="true"/>
<serviceDebug includeExceptionDetailInFaults="true"/>

```

My client application for WM 6.0.

Code Snippet

```

using System;
using System.Collections.Generic;
using System.Collections.ObjectModel;
using System.Text;
using Microsoft.Synchronization;
using Microsoft.Synchronization.Data;
using Microsoft.Synchronization.Data.SqlServerCe;
using System.Data;
using System.Data.Common;
using System.Data.SqlClient;
using System.Data.SqlServerCe;
using System.IO;
namespace ocsClient_device
{
class Program
{
static void Main(string[] args)

```

```

{
// test the basic method to ensure the connection, basic call worked on
devices
ServiceClient svc = new ServiceClient();
string testString = svc.GetData(3);
OcsClient client = new OcsClient();
client.SetupAgent();
client.Sync();
}
}
class OcsClient
{
const string clientdb = "test.sdf";
const string connString = "data source = test.sdf";
SyncAgent sa;
const string tableName = "table1";
public void SetupAgent()
{
SqlCeClientSyncProvider clientProvider = new
SqlCeClientSyncProvider(connString, true);
ServiceClient svc = new ServiceClient();
ServerSyncProvider serverProvider = new ServerSyncProviderProxy(svc);
sa = new SyncAgent();
sa.Configuration.SyncTables.Add(tableName, TableCreationOption.CreateNewTableO
rFail, Microsoft.Synchronization.Data.SyncDirection.DownloadOnly);
sa.LocalProvider = clientProvider;
sa.RemoteProvider = serverProvider;
}
public void Sync()
{
try
{
SyncStatistics stats = sa.Synchronize();
}
catch (Exception e)
{
Console.WriteLine(e.ToString());
}
}
}
}
}

```

this is the modified service.cs on client side, please change the YourServerName with the proper server name.

Code Snippet

```

//-----
---
// <auto-generated>
// This code was generated by a tool.

```

```

// Runtime Version:2.0.50727.3051
//
// Changes to this file may cause incorrect behavior and will be lost if
// the code is regenerated.
// </auto-generated>
//-----
---
using System.Data;
using Microsoft.Synchronization;
using Microsoft.Synchronization.Data;
using Microsoft.Synchronization.Data.SqlServerCe;

[System.CodeDom.Compiler.GeneratedCodeAttribute("System.ServiceModel",
"3.0.0.0")]
public interface IService
{
    string GetData(int value);
    CompositeType GetDataUsingDataContract(CompositeType composite);
    SyncServerInfo GetServerInfo(SyncSession syncSession);
    SyncContext ApplyChanges(SyncGroupMetadata groupMetadata, System.Data.DataSet
dataSet, SyncSession syncSession);
    SyncContext GetChanges(SyncGroupMetadata groupMetadata, SyncSession
syncSession);
    SyncSchema GetSchema(string[] tableNames, SyncSession syncSession);
}
[System.Diagnostics.DebuggerStepThroughAttribute()]
[System.CodeDom.Compiler.GeneratedCodeAttribute("System.ServiceModel",
"3.0.0.0")]
[System.Xml.Serialization.XmlRootAttribute(ElementName="GetData",
Namespace="http://tempuri.org/")]
public partial class GetDataRequest
{
    [System.Xml.Serialization.XmlElementAttribute(Namespace="http://tempuri.org/"
, Order=0)]
    public int value;
    public GetDataRequest()
    {
    }
    public GetDataRequest(int value)
    {
        this.value = value;
    }
}
[System.Diagnostics.DebuggerStepThroughAttribute()]
[System.CodeDom.Compiler.GeneratedCodeAttribute("System.ServiceModel",
"3.0.0.0")]
[System.Xml.Serialization.XmlRootAttribute(ElementName="GetDataResponse",
Namespace="http://tempuri.org/")]
public partial class GetDataResponse
{

```

```

[System.Xml.Serialization.XmlElementAttribute (Namespace="http://tempuri.org/"
, Order=0)]
public string GetDataResult;
public GetDataResponse()
{
}
public GetDataResponse(string GetDataResult)
{
this.GetDataResult = GetDataResult;
}
}
/// <remarks/>
[System.CodeDom.Compiler.GeneratedCodeAttribute ("NetCFSvcUtil", "3.5.0.0")]
[System.SerializableAttribute()]
[System.Diagnostics.DebuggerStepThroughAttribute()]
[System.ComponentModel.DesignerCategoryAttribute ("code")]
[System.Xml.Serialization.XmlTypeAttribute (Namespace="http://tempuri.org/") ]
public partial class CompositeType
{
private bool boolValueField;
private string stringValueField;
/// <remarks/>
[System.Xml.Serialization.XmlElementAttribute (Order=0)]
public bool BoolValue
{
get
{
return this.boolValueField;
}
set
{
this.boolValueField = value;
}
}
/// <remarks/>
[System.Xml.Serialization.XmlElementAttribute (Order=1)]
public string StringValue
{
get
{
return this.stringValueField;
}
set
{
this.stringValueField = value;
}
}
}
[System.Diagnostics.DebuggerStepThroughAttribute()]
[System.CodeDom.Compiler.GeneratedCodeAttribute ("System.ServiceModel",
"3.0.0.0")]

```



```

[System.Xml.Serialization.XmlRootAttribute (ElementName="GetDataUsingDataContract", Namespace="http://tempuri.org/")]
public partial class GetDataUsingDataContractRequest
{
[System.Xml.Serialization.XmlElementAttribute (Namespace="http://tempuri.org/", Order=0)]
public CompositeType composite;
public GetDataUsingDataContractRequest ()
{
}
public GetDataUsingDataContractRequest (CompositeType composite)
{
this.composite = composite;
}
}
[System.Diagnostics.DebuggerStepThroughAttribute ()]
[System.CodeDom.Compiler.GeneratedCodeAttribute ("System.ServiceModel", "3.0.0.0")]
[System.Xml.Serialization.XmlRootAttribute (ElementName="GetDataUsingDataContractResponse", Namespace="http://tempuri.org/")]
public partial class GetDataUsingDataContractResponse
{
[System.Xml.Serialization.XmlElementAttribute (Namespace="http://tempuri.org/", Order=0)]
public CompositeType GetDataUsingDataContractResult;
public GetDataUsingDataContractResponse ()
{
}
public GetDataUsingDataContractResponse (CompositeType GetDataUsingDataContractResult)
{
this.GetDataUsingDataContractResult = GetDataUsingDataContractResult;
}
}
[System.Diagnostics.DebuggerStepThroughAttribute ()]
[System.CodeDom.Compiler.GeneratedCodeAttribute ("System.ServiceModel", "3.0.0.0")]
[System.Xml.Serialization.XmlRootAttribute (ElementName="GetServerInfo", Namespace="http://tempuri.org/")]
public partial class GetServerInfoRequest
{
[System.Xml.Serialization.XmlElementAttribute (Namespace="http://tempuri.org/", Order=0)]
public SyncSession syncSession;
public GetServerInfoRequest ()
{
}
public GetServerInfoRequest (SyncSession syncSession)
{
this.syncSession = syncSession;
}
}

```

```

}
[System.Diagnostics.DebuggerStepThroughAttribute()]
[System.CodeDom.Compiler.GeneratedCodeAttribute("System.ServiceModel",
"3.0.0.0")]
[System.Xml.Serialization.XmlRootAttribute(ElementName="GetServerInfoResponse",
Namespace="http://tempuri.org/")]
public partial class GetServerInfoResponse
{
[System.Xml.Serialization.XmlElementAttribute(Namespace="http://tempuri.org/",
Order=0)]
public SyncServerInfo GetServerInfoResult;
public GetServerInfoResponse()
{
}
public GetServerInfoResponse(SyncServerInfo GetServerInfoResult)
{
this.GetServerInfoResult = GetServerInfoResult;
}
}
[System.Diagnostics.DebuggerStepThroughAttribute()]
[System.CodeDom.Compiler.GeneratedCodeAttribute("System.ServiceModel",
"3.0.0.0")]
[System.Xml.Serialization.XmlRootAttribute(ElementName="ApplyChanges",
Namespace="http://tempuri.org/")]
public partial class ApplyChangesRequest
{
[System.Xml.Serialization.XmlElementAttribute(Namespace="http://tempuri.org/",
Order=0)]
public SyncGroupMetadata groupMetadata;
[System.Xml.Serialization.XmlElementAttribute(Namespace="http://tempuri.org/",
Order=1)]
public System.Data.DataSet dataSet;
[System.Xml.Serialization.XmlElementAttribute(Namespace="http://tempuri.org/",
Order=2)]
public SyncSession syncSession;
public ApplyChangesRequest()
{
}
public ApplyChangesRequest(SyncGroupMetadata groupMetadata,
System.Data.DataSet dataSet, SyncSession syncSession)
{
this.groupMetadata = groupMetadata;
this.dataSet = dataSet;
this.syncSession = syncSession;
}
}
[System.Diagnostics.DebuggerStepThroughAttribute()]
[System.CodeDom.Compiler.GeneratedCodeAttribute("System.ServiceModel",
"3.0.0.0")]
[System.Xml.Serialization.XmlRootAttribute(ElementName="ApplyChangesResponse",
Namespace="http://tempuri.org/")]

```

```

public partial class ApplyChangesResponse
{
[System.Xml.Serialization.XmlElementAttribute (Namespace="http://tempuri.org/",
, Order=0)]
public SyncContext ApplyChangesResult;
public ApplyChangesResponse()
{
}
public ApplyChangesResponse(SyncContext ApplyChangesResult)
{
this.ApplyChangesResult = ApplyChangesResult;
}
}
[System.Diagnostics.DebuggerStepThroughAttribute()]
[System.CodeDom.Compiler.GeneratedCodeAttribute("System.ServiceModel",
"3.0.0.0")]
[System.Xml.Serialization.XmlRootAttribute (ElementName="GetChanges",
Namespace="http://tempuri.org/")]
public partial class GetChangesRequest
{
[System.Xml.Serialization.XmlElementAttribute (Namespace="http://tempuri.org/",
, Order=0)]
public SyncGroupMetadata groupMetadata;
[System.Xml.Serialization.XmlElementAttribute (Namespace="http://tempuri.org/",
, Order=1)]
public SyncSession syncSession;
public GetChangesRequest()
{
}
public GetChangesRequest(SyncGroupMetadata groupMetadata, SyncSession
syncSession)
{
this.groupMetadata = groupMetadata;
this.syncSession = syncSession;
}
}
[System.Diagnostics.DebuggerStepThroughAttribute()]
[System.CodeDom.Compiler.GeneratedCodeAttribute("System.ServiceModel",
"3.0.0.0")]
[System.Xml.Serialization.XmlRootAttribute (ElementName="GetChangesResponse",
Namespace="http://tempuri.org/")]
public partial class GetChangesResponse
{
[System.Xml.Serialization.XmlElementAttribute (Namespace="http://tempuri.org/",
, Order=0)]
public SyncContext GetChangesResult;
public GetChangesResponse()
{
}
public GetChangesResponse(SyncContext GetChangesResult)
{
}
}

```

```

this.GetChangesResult = GetChangesResult;
}
}
[System.Diagnostics.DebuggerStepThroughAttribute()]
[System.CodeDom.Compiler.GeneratedCodeAttribute("System.ServiceModel",
"3.0.0.0")]
[System.Xml.Serialization.XmlRootAttribute(ElementName="GetSchema",
Namespace="http://tempuri.org/")]
public partial class GetSchemaRequest
{
[System.Xml.Serialization.XmlArrayAttribute(Namespace="http://tempuri.org/",
Order=0)]
public string[] tableNames;
[System.Xml.Serialization.XmlElementAttribute(Namespace="http://tempuri.org/"
, Order=1)]
public SyncSession syncSession;
public GetSchemaRequest()
{
}
public GetSchemaRequest(string[] tableNames, SyncSession syncSession)
{
this.tableNames = tableNames;
this.syncSession = syncSession;
}
}
[System.Diagnostics.DebuggerStepThroughAttribute()]
[System.CodeDom.Compiler.GeneratedCodeAttribute("System.ServiceModel",
"3.0.0.0")]
[System.Xml.Serialization.XmlRootAttribute(ElementName="GetSchemaResponse",
Namespace="http://tempuri.org/")]
public partial class GetSchemaResponse
{
[System.Xml.Serialization.XmlElementAttribute(Namespace="http://tempuri.org/"
, Order=0)]
public SyncSchema GetSchemaResult;
public GetSchemaResponse()
{
}
public GetSchemaResponse(SyncSchema GetSchemaResult)
{
this.GetSchemaResult = GetSchemaResult;
}
}
[System.Diagnostics.DebuggerStepThroughAttribute()]
[System.CodeDom.Compiler.GeneratedCodeAttribute("System.ServiceModel",
"3.0.0.0")]
public partial class ServiceClient :
Microsoft.Tools.ServiceModel.CFClientBase<IService>, IService
{

```

```

public static System.ServiceModel.EndpointAddress EndpointAddress = new
System.ServiceModel.EndpointAddress ("http://<yourServerName>/WCFTTest/Service.svc");
public ServiceClient() :
this(CreateDefaultBinding(), EndpointAddress)
{
}
public ServiceClient(System.ServiceModel.Channels.Binding binding,
System.ServiceModel.EndpointAddress remoteAddress) :
base(binding, remoteAddress)
{
addProtectionRequirements(binding);
}
private GetDataResponse GetData(GetDataRequest request)
{
CFInvokeInfo info = new CFInvokeInfo();
info.Action = "http://tempuri.org/IService/GetData";
info.RequestIsWrapped = true;
info.ReplyAction = "http://tempuri.org/IService/GetDataResponse";
info.ResponseIsWrapped = true;
GetDataResponse retVal = base.Invoke<GetDataRequest, GetDataResponse>(info,
request);
return retVal;
}
public string GetData(int value)
{
GetDataRequest request = new GetDataRequest(value);
GetDataResponse response = this.GetData(request);
return response.GetDataResult;
}
private GetDataUsingDataContractResponse
GetDataUsingDataContract(GetDataUsingDataContractRequest request)
{
CFInvokeInfo info = new CFInvokeInfo();
info.Action = "http://tempuri.org/IService/GetDataUsingDataContract";
info.RequestIsWrapped = true;
info.ReplyAction =
"http://tempuri.org/IService/GetDataUsingDataContractResponse";
info.ResponseIsWrapped = true;
GetDataUsingDataContractResponse retVal =
base.Invoke<GetDataUsingDataContractRequest,
GetDataUsingDataContractResponse>(info, request);
return retVal;
}
public CompositeType GetDataUsingDataContract(CompositeType composite)
{
GetDataUsingDataContractRequest request = new
GetDataUsingDataContractRequest(composite);
GetDataUsingDataContractResponse response =
this.GetDataUsingDataContract(request);
return response.GetDataUsingDataContractResult;
}

```

```

}
private GetServerInfoResponse GetServerInfo(GetServerInfoRequest request)
{
    CFInvokeInfo info = new CFInvokeInfo();
    info.Action = "http://tempuri.org/IService/GetServerInfo";
    info.RequestIsWrapped = true;
    info.ReplyAction = "http://tempuri.org/IService/GetServerInfoResponse";
    info.ResponseIsWrapped = true;
    GetServerInfoResponse retVal = base.Invoke<GetServerInfoRequest,
    GetServerInfoResponse>(info, request);
    return retVal;
}
public SyncServerInfo GetServerInfo(SyncSession syncSession)
{
    GetServerInfoRequest request = new GetServerInfoRequest(syncSession);
    GetServerInfoResponse response = this.GetServerInfo(request);
    return response.GetServerInfoResult;
}
private ApplyChangesResponse ApplyChanges(ApplyChangesRequest request)
{
    CFInvokeInfo info = new CFInvokeInfo();
    info.Action = "http://tempuri.org/IService/ApplyChanges";
    info.RequestIsWrapped = true;
    info.ReplyAction = "http://tempuri.org/IService/ApplyChangesResponse";
    info.ResponseIsWrapped = true;
    ApplyChangesResponse retVal = base.Invoke<ApplyChangesRequest,
    ApplyChangesResponse>(info, request);
    return retVal;
}
public SyncContext ApplyChanges(SyncGroupMetadata groupMetadata,
System.Data.DataSet dataSet, SyncSession syncSession)
{
    ApplyChangesRequest request = new ApplyChangesRequest(groupMetadata, dataSet,
syncSession);
    ApplyChangesResponse response = this.ApplyChanges(request);
    return response.ApplyChangesResult;
}
private GetChangesResponse GetChanges(GetChangesRequest request)
{
    CFInvokeInfo info = new CFInvokeInfo();
    info.Action = "http://tempuri.org/IService/GetChanges";
    info.RequestIsWrapped = true;
    info.ReplyAction = "http://tempuri.org/IService/GetChangesResponse";
    info.ResponseIsWrapped = true;
    GetChangesResponse retVal = base.Invoke<GetChangesRequest,
    GetChangesResponse>(info, request);
    return retVal;
}
public SyncContext GetChanges(SyncGroupMetadata groupMetadata, SyncSession
syncSession)
{

```

```

GetChangesRequest request = new GetChangesRequest (groupMetadata,
syncSession);
GetChangesResponse response = this.GetChanges (request);
return response.GetChangesResult;
}
private GetSchemaResponse GetSchema (GetSchemaRequest request)
{
CFInvokeInfo info = new CFInvokeInfo ();
info.Action = "http://tempuri.org/IService/GetSchema";
info.RequestIsWrapped = true;
info.ReplyAction = "http://tempuri.org/IService/GetSchemaResponse";
info.ResponseIsWrapped = true;
GetSchemaResponse retVal = base.Invoke<GetSchemaRequest,
GetSchemaResponse>(info, request);
return retVal;
}
public SyncSchema GetSchema (string[] tableNames, SyncSession syncSession)
{
GetSchemaRequest request = new GetSchemaRequest (tableNames, syncSession);
GetSchemaResponse response = this.GetSchema (request);
return response.GetSchemaResult;
}
public static System.ServiceModel.Channels.Binding CreateDefaultBinding ()
{
System.ServiceModel.Channels.CustomBinding binding = new
System.ServiceModel.Channels.CustomBinding ();
binding.Elements.Add (new
System.ServiceModel.Channels.TextMessageEncodingBindingElement (System.Service
Model.Channels.MessageVersion.Soap11, System.Text.Encoding.UTF8));
binding.Elements.Add (new
System.ServiceModel.Channels.HttpTransportBindingElement ());
return binding;
}
private void addProtectionRequirements (System.ServiceModel.Channels.Binding
binding)
{
if ((IsSecureMessageBinding (binding) == false))
{
return;
}
System.ServiceModel.Security.ChannelProtectionRequirements cpr = new
System.ServiceModel.Security.ChannelProtectionRequirements ();
ApplyProtection ("http://tempuri.org/IService/GetData",
cpr.IncomingSignatureParts, true);
ApplyProtection ("http://tempuri.org/IService/GetData",
cpr.IncomingEncryptionParts, true);
ApplyProtection ("http://tempuri.org/IService/GetDataUsingDataContract",
cpr.IncomingSignatureParts, true);
ApplyProtection ("http://tempuri.org/IService/GetDataUsingDataContract",
cpr.IncomingEncryptionParts, true);
}

```

```

ApplyProtection("http://tempuri.org/IService/GetServerInfo",
cpr.IncomingSignatureParts, true);
ApplyProtection("http://tempuri.org/IService/GetServerInfo",
cpr.IncomingEncryptionParts, true);
ApplyProtection("http://tempuri.org/IService/ApplyChanges",
cpr.IncomingSignatureParts, true);
ApplyProtection("http://tempuri.org/IService/ApplyChanges",
cpr.IncomingEncryptionParts, true);
ApplyProtection("http://tempuri.org/IService/GetChanges",
cpr.IncomingSignatureParts, true);
ApplyProtection("http://tempuri.org/IService/GetChanges",
cpr.IncomingEncryptionParts, true);
ApplyProtection("http://tempuri.org/IService/GetSchema",
cpr.IncomingSignatureParts, true);
ApplyProtection("http://tempuri.org/IService/GetSchema",
cpr.IncomingEncryptionParts, true);
if ((binding.MessageVersion.Addressing ==
System.ServiceModel.Channels.AddressingVersion.None))
{
ApplyProtection("*", cpr.OutgoingSignatureParts, true);
ApplyProtection("*", cpr.OutgoingEncryptionParts, true);
}
else
{
ApplyProtection("http://tempuri.org/IService/GetDataResponse",
cpr.OutgoingSignatureParts, true);
ApplyProtection("http://tempuri.org/IService/GetDataResponse",
cpr.OutgoingEncryptionParts, true);
ApplyProtection("http://tempuri.org/IService/GetDataUsingDataContractResponse",
cpr.OutgoingSignatureParts, true);
ApplyProtection("http://tempuri.org/IService/GetDataUsingDataContractResponse",
cpr.OutgoingEncryptionParts, true);
ApplyProtection("http://tempuri.org/IService/GetServerInfoResponse",
cpr.OutgoingSignatureParts, true);
ApplyProtection("http://tempuri.org/IService/GetServerInfoResponse",
cpr.OutgoingEncryptionParts, true);
ApplyProtection("http://tempuri.org/IService/ApplyChangesResponse",
cpr.OutgoingSignatureParts, true);
ApplyProtection("http://tempuri.org/IService/ApplyChangesResponse",
cpr.OutgoingEncryptionParts, true);
ApplyProtection("http://tempuri.org/IService/GetChangesResponse",
cpr.OutgoingSignatureParts, true);
ApplyProtection("http://tempuri.org/IService/GetChangesResponse",
cpr.OutgoingEncryptionParts, true);
ApplyProtection("http://tempuri.org/IService/GetSchemaResponse",
cpr.OutgoingSignatureParts, true);
ApplyProtection("http://tempuri.org/IService/GetSchemaResponse",
cpr.OutgoingEncryptionParts, true);
}
this.Parameters.Add(cpr);
}
}

```