

# INTRODUKTION – WCFSERVICES

## INLEDNING

Det här dokumentet är till för att ge en översikt över hur WCF Services används från .NET applikationer.

Då flertalet av de termer och koncept som förklaras och används i instruktionerna gör sig bäst på engelska varnas känsliga läsare för en stor dos av ”svengelska” i detta dokument, det var mer eller mindre ofrånkomligt.

## INNEHÅLL

Inledning .....	1
WCF (Windows Communication Foundation) .....	2
Lägg till en service reference .....	2
Använda Servicen .....	3
Vanliga fel.....	3

## WCF (WINDOWS COMMUNICATION FOUNDATION)

Windows Communication Foundation (WCF) är ett ramverk för att bygga tjänsteorienterade program. Med WCF kan du skicka data som asynkrona meddelanden från en tjänstslutpunkt till en annan. En tjänstslutpunkt kan vara en del av en ständigt tillgänglig tjänst som är hostas på IIS (Internet Information Services), eller så kan den vara en tjänst som finns i ett program. En slutpunkt kan vara en klient för en tjänst som begär data från en tjänstslutpunkt. Meddelandena kan vara så enkla som ett enstaka tecken eller ord som skickas som XML, eller så komplexa som en ström av binära data. Några exempelscenarier är:

- En säker tjänst för att bearbeta affärstransaktioner.
- En tjänst som tillhandahåller aktuella data till andra, till exempel en trafik rapport eller annan övervakningstjänst.
- En chatttjänst som gör att två personer kan kommunicera eller utbyta data i realtid.
- Ett Instrumentpanels program som avsöker en eller flera tjänster för data och presenterar det i en logisk presentation.
- Exponera ett arbetsflöde som implementerats med Windows Workflow Foundation som en WCF-tjänst.

Medan att skapa sådana program var möjligt innan förekomsten av WCF, dock gör WCF utvecklingen av slutpunkter lättare än någonsin. Sammanfattningsvis är WCF utformat för att erbjuda en hanterbar metod för att skapa webbtjänster och webbtjänst-klienter.

### Lägg till en service reference

När man har skapat sin service måste man lägga till den i klienten för att kunna använda den. En WCF service läggs till i klienten genom att man lägger till en referens till servicen i klienten. För att lägga till en servicereferens högerklickar man på klientens referensmapp (References) och väljer "Add Service Reference". Om det är en extern service som man vill lägga till så behöver man skriva in adressen till servicen i det fönster som dyker upp och sedan trycka på "Go". Om det är en service som finns i samma solution som klienten så kan man klicka på "Discover". Om man har skrivit sin service korrekt med ett interface som har [ServiceContract] och [OperationContract] taggar så bör servicen dyka upp i fönstret. Nästa steg är att ge servicen ett namespace. Man brukar vanligtvis antingen endast skriva in det namn som servicen har alternativt servicensnamn + Reference för att särskilja på servicen och referensen.

## Använda Servicen

I de klasser som ni har tänkt använda er utav information från servicen måste ni nu använda er utav servicereferensen. Detta görs genom att skriva using

NAMN\_PÅ\_SOLUTION.SERVICEREFERENSENS\_NAMESPACE. Detta kan exempelvis se ut på följande vis:  
`using Lab3.ServiceReference;`

När man har lagt till detta så kan man deklarera och instansiera en klient som man använder för att kommunicera med servicen. I exemplet nedan så heter servicen ExampleService och för att instansiera en klient så skriver man ExampleServiceClient

variabelnamn\_på\_klientobjekt = new ExampleServiceClient(). Genom detta objekt kan man sedan kalla på de metoder som servicen exponerar (de metoder som är deklarerade med OperationContract i servicens interface).

För att inte servicen ska bli överbelastad så bör man stänga kopplingen till servicen när man har kallat på de metoder som man tänkt kalla på från servicen. Detta görs genom att kalla på klientobjektets Close metod ( variabelnamn\_på\_klientobjekt.Close(); )

## Vanliga fel

Ett vanligt fel som kan uppstå är att projektet inte kan lägga till servicereferensen. Detta fel uppstår oftast om man inte har byggt sin service. Lösningen på detta är att köra Clean Solution följt av Build Solution. Detta kan göras antingen genom att högerklicka på Solution i Solution Explorer och välja Clean Solution följt av Build Solution. Clean/Build Solution finns även under Build menyn uppe i Visual Studio. Man kan även köra Clean och Build endast på servicen genom att högerklicka på serviceprojektet och välja Clean och sedan Build. Notera att man endast kan göra en ny Build om man inte har några fel i sin kod.

Om man gör förändringar i servicen (såsom lägger till nya metoder eller ändrar metodsignaturer) efter att man lagt till en referens till servicen i klienten så måste man uppdatera servicereferensen hos klienten. Detta gör man genom att högerklicka på servicereferensen och välja "Update Service Reference". Servicereferensen finner man under "Connected Services" i klient-projektet.

Om ni använder er utav skolans datorer/fjärrskrivbord så kan det hända att ni kopplar till någon annans service. Om detta skulle ske så behöver ni ändra port som er service lyssnar på. Detta görs genom att högerklicka på ert serviceprojekt och välja Properties (alternativt markera serviceprojektet och trycka Alt+Enter). I fönstret som dyker upp går ni in under Web och ändra portnumret Project URL. När ni har gjort detta måste ni sedan trycka på Create Virtual Directory. Därefter måste nu köra Clean och Build på projektet och sedan ta bort servicereferensen ifrån klientprojektet och lägga till den på nytt.

Om ni får ett `CommunicationException` med meddelandet "The maximum message size quota for incoming messages (65536) has been exceeded." så behöver ni sätta `maxReceivedMessageSize` i `App.config` för klienten. Detta ska läggas till i `binding` taggen under `basicHttpBinding` taggen. Om `binding` taggen inte finns så skapar ni den. Sätt sedan `maxReceivedMessageSize` till 2147483647 genom att se till att `App.config` innehåller:

```
<bindings>
  <basicHttpBinding>
    <binding name="BasicHttpBinding_IExampleService" maxReceivedMessageSize="2147483647"/>
  </basicHttpBinding>
</bindings>
```

Det kan även vara bra att använda sig utav try-catch för att hantera vissa felmeddelanden. T.ex. om er service inte lyckas omvandla JSON till XML så ska programmet inte krascha utan ett felmeddelande ska returneras till klienten istället för den omvandlade XML:en. Try-catch används på följande vis:

```
public XElement FilterByNode(string s)
{
    try
    {
        //Kod som skall köras men som kan kasta ett exception
    }
    catch (Exception e)
    {
        return new XElement("Error", e.Message);
    }
}
```

Tanken är att ert program ska vara stabilt och inte krascha oavsett vad en användare gör. Så se till att hantera era felmeddelanden och även att ni ger användaren information om vad som gått fel.