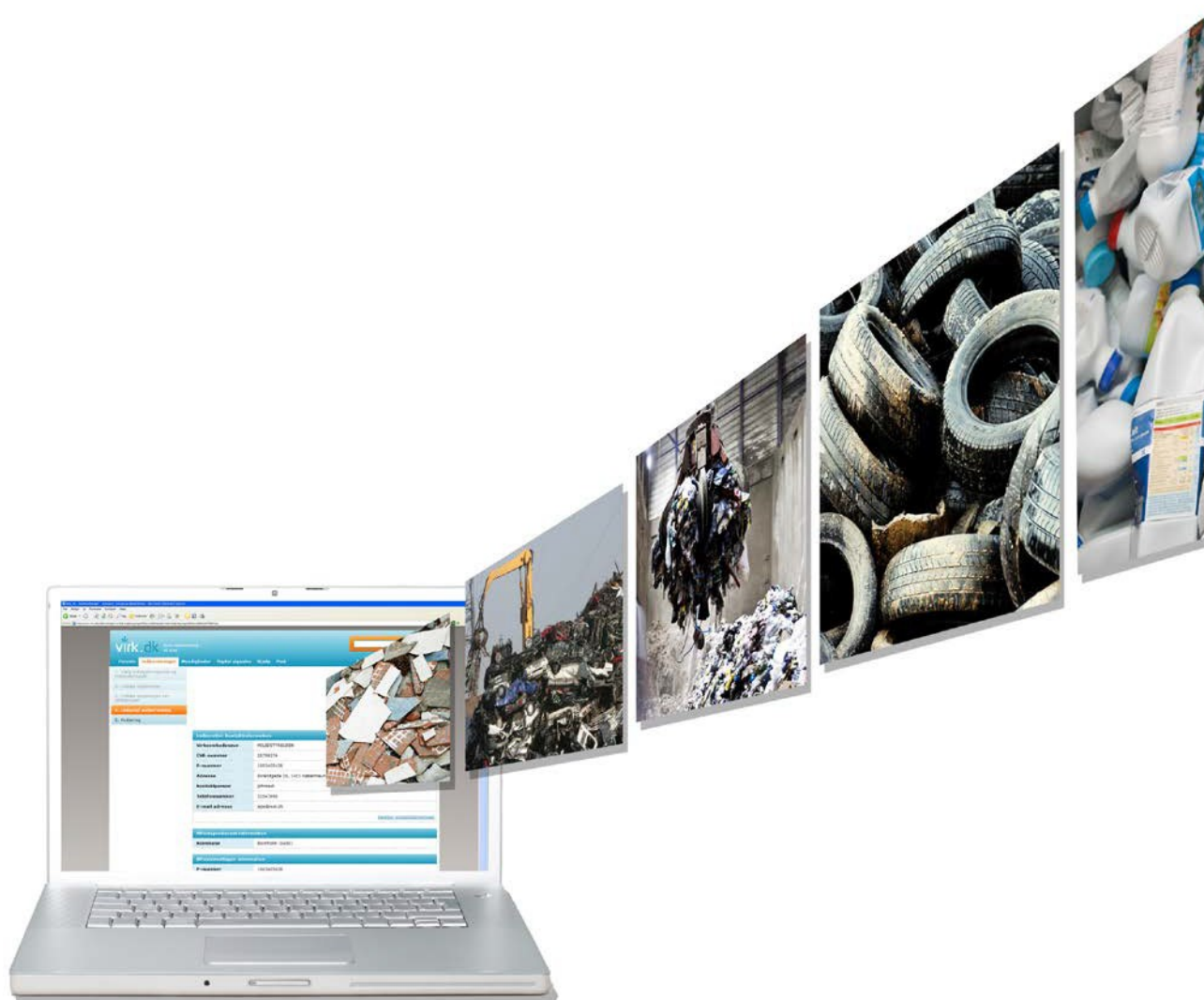


Affaldsdatasystem

Vejledning supplement i system-til-system integration for .Net brugere



Dokument version: 2.3
ADS version: 2.4



Miljøministeriet
Miljøstyrelsen

Henvendelse vedrørende affald:
Miljøstyrelsen, Affaldsdatasystem
Haraldsgade 53
2100 København Ø
Tlf. 72 54 81 81
affaldsdatasystem@mst.dk

Yderligere information:
www.mst.dk

Forsidefoto: Colourbox

Indhold

1	INTRODUKTION	4
1.1	REFERENCER	4
2	FORUDSÆTNINGER	5
2.1	MILJØER	5
2.2	ADGANG	5
3	EKSEMPLER	7
3.1	OPSÆTNING OG BRUG AF SERVICE REFERENCE.....	7
3.2	OPSÆTNING OG BRUG AF SERVICE CHANNEL.....	13
4	CHANGE LOG	15

1 Introduktion

Dette dokument er en vejledning til at komme i gang med at bruge webservicen der muliggør system-system indberetning for Affaldsdatasystem (ADS). Vejledningen er skrevet for brugere af Visual Studio/.Net og beskriver skridt for skridt, hvorledes man kan opnå adgang til, og anvende servicen. Der gives to eksempler, der viser til forskellige tilgange til at gøre dette.

Webservicen og betingelser for brug, er beskrevet i detaljer i [1]. ***Dette dokument skal kun ses som et supplement til den fulde vejledning.*** Altså en ekstra vejledning der gælder specifikt for aftagere af system-system webservicen, der implementerer deres løsning ved hjælp af .Net.

1.1 Referencer

[1] "Vejledning til system til system integration", version 2.7, NNIT.

2 Forudsætninger

Detaljer om forudsætninger og miljøer til brug af webservicen er beskrevet i [1]. Her opsummeres kort de forudsætninger der skal være opfyldt for at komme i gang med brug af servicen.

2.1 Miljøer

Affaldsdatasystemet har et Uddannelses- og et PROD miljø. Uddannelsesmiljøet skal benyttes til test og kvalitetssikring af serviceaftagers integration med affaldsdatasystemet. PROD er affaldsdatasystemet produktionsmiljøet.

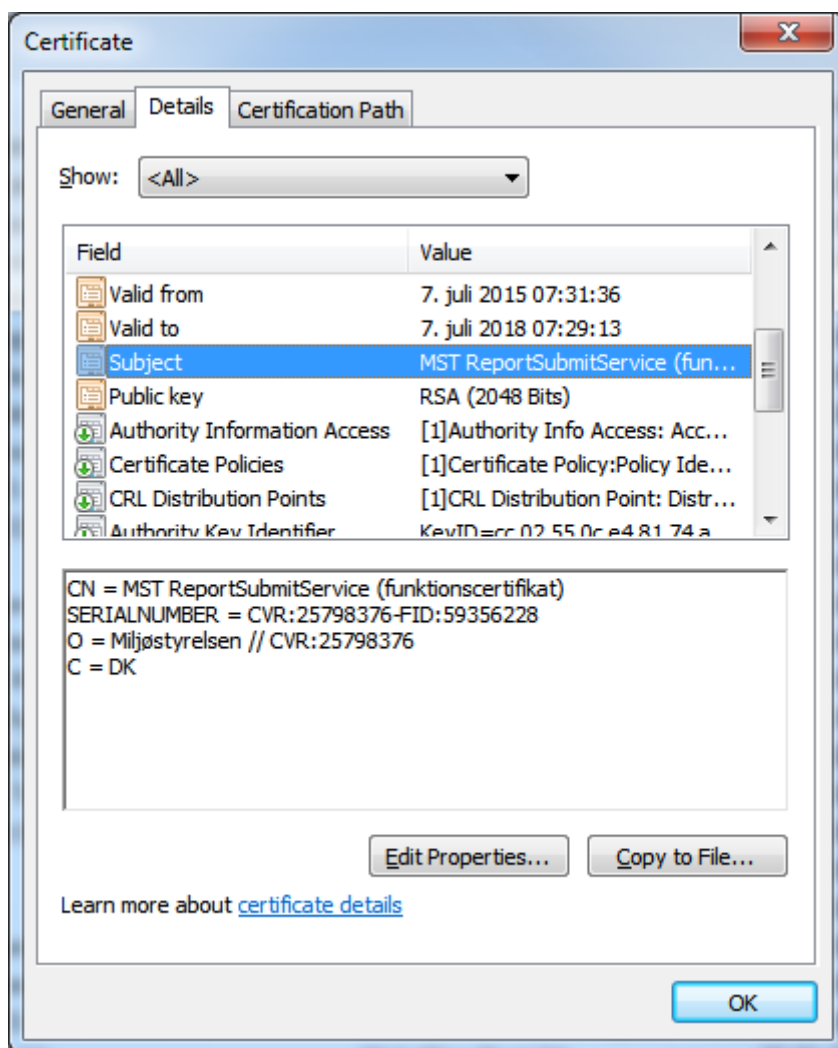
UDD miljø	URL
Web service	https://udd.ws.ads.mst.dk/ReportSubmitService.svc
Web service description	https://udd.ws.ads.mst.dk/AdsReporting.wsdl
Affaldsdatasystemets hjemmeside	https://udd.ads.mst.dk
Placering af skemaer	https://udd.ws.ads.mst.dk/schemas/
Certifikat	OCES II funktions certifikat. Udsteder skal være: TRUST2408 OCES CA II

PROD miljø	URL
Web service	https://www.ws.ads.mst.dk/ReportSubmitService.svc
Web service description	https://ws.ads.mst.dk/AdsReporting.wsdl
Affaldsdatasystemets hjemmeside	https://www.ads.mst.dk
Placering af skemaer	https://ws.ads.mst.dk/schemas/
Certifikat	OCES II virksomheds- eller funktionscertifikat. Udsteder skal være: TRUST2408 OCES CA II

2.2 Adgang

For at opnå adgang, skal følgende være opfyldt:

- Virksomheden skal anskaffe et virksomhedscertifikat (VOCES) eller alternativt, et funktionscertifikat (FOCES). Certifikater kan rekvireres hos DanID: <https://danid.dk/export/sites/dk.danid.oc/da/erhverv/>
- Derefter skal virksomheden registreres som systembruger i Affaldsdatasystemet af Affaldssekretariatet (72 54 81 81). Registreringen sker per CVR-nummer, som skal matche det CVR nummer, der er angivet i aftagerens certifikats subject felt:



Bemærk at *hele* webservice sitet kræver autentificering med et validt certifikat. Det gælder også tilgang til webservice beskrivelse og skemaer. Vil man derfor teste om et rekvireret certifikat er gyldigt for webservicen, kan man forsøge at tilgå WSDL filen, f.eks. i en browser, og autentificerer sig med sit certifikat. Er certifikatet ikke validt, eller ikke den rigtige type, vil ADS svare med HTTP status kode "403 Forbiden".

3 Eksempler

Der findes mange måder at opnå adgang til og anvende en webservice. Disse vil også variere afhængigt af hvilken version af VS og .Net frameworket man anvender.

Disse eksempler anvender VS2015, .Net 4.5.2, og C# version 6.0, i et console projekt.

Nedenstående afsnit viser eksempler på to forskellige fremgangsmåder:

- At anvende en "service reference". En fremgangsmåde der lader VS autogenerere klasser og metoder, der kan anvendes til at kalde webservicen.
- At definere et endpoint, binding og en request SOAP besked, der anvendes ved at oprette en såkaldt channel til webservicen, og kalde den derigennem. Altså en fremgangsmåde der er "tættere" på transport laget af et servicekald. En fordel ved denne fremgangsmåde, kan være at det tillader at definere indholdet af den sendte request SOAP besked, som ren XML.

3.1 Opsætning og brug af service reference

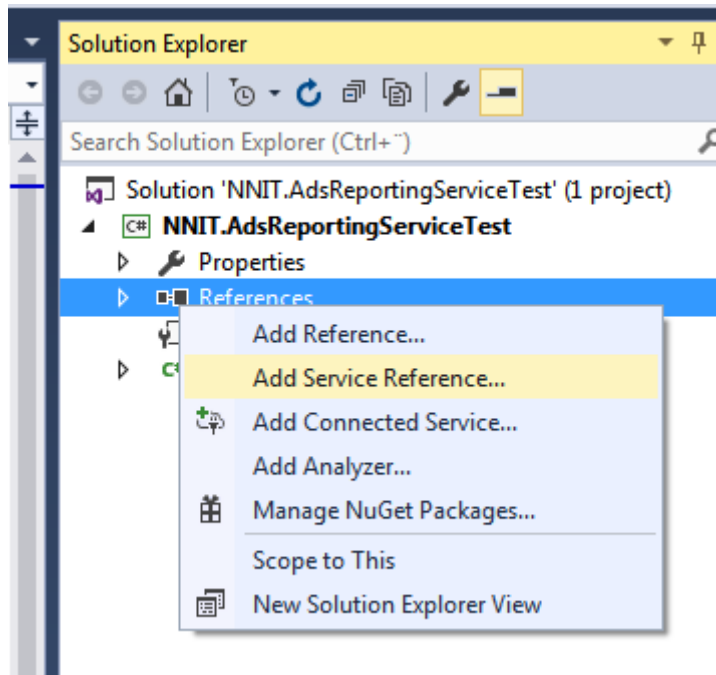
Dette eksempel viser skridt for skridt, hvordan man i et nyoprettet Visual Studio projekt kan opnå adgang til system-system webservicen, og anvende den til at sende en indberetning, ved at anvende en "service reference". Udtræk alle filer fra AdsReporting.zip filen som du kan hente samme sted som denne vejledning. De skal ligge i en struktur som denne:



Name	Date modified	Type	Size
Schemas	04-04-2016 14:59	File folder	
AdsReporting.wsdl	18-12-2015 07:03	Web Services Description Language	11 KB

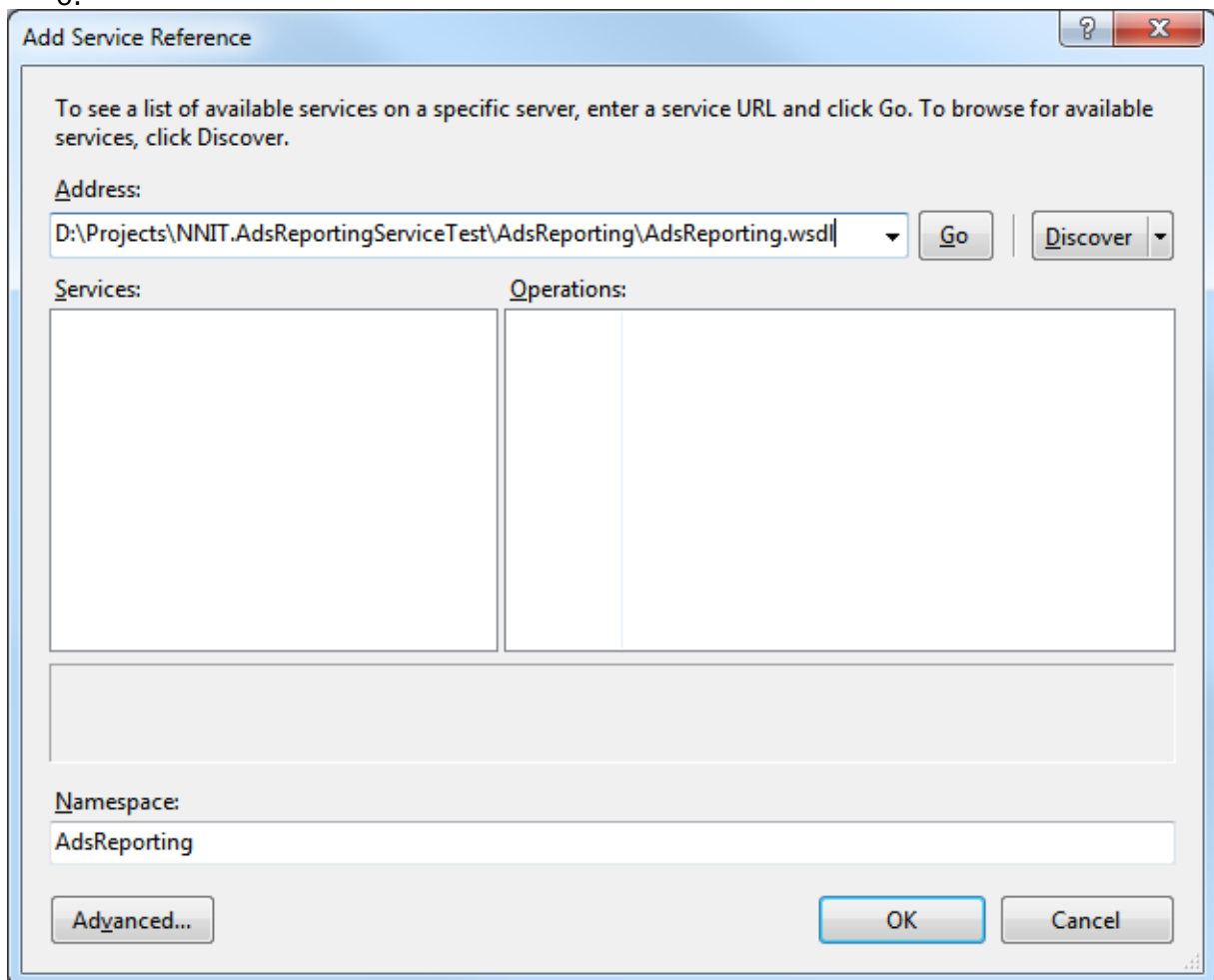
1. Installerer det anskaffede certifikat et passende sted i dit certificate store, f.eks. "Current user/LocalMachine".
2. Verificer at du kan oprette en SSL forbindelse til webservice sitet, ved at åbne stien til WSDL filen <https://udd.ws.ads.mst.dk/AdsReporting.wsdl> i en browser. Du bør blive prompted for et certifikat og password.
3. Opret et nyt Visual Studio projekt og tilføj en ny service reference, her er valgt et windows console program:

4.

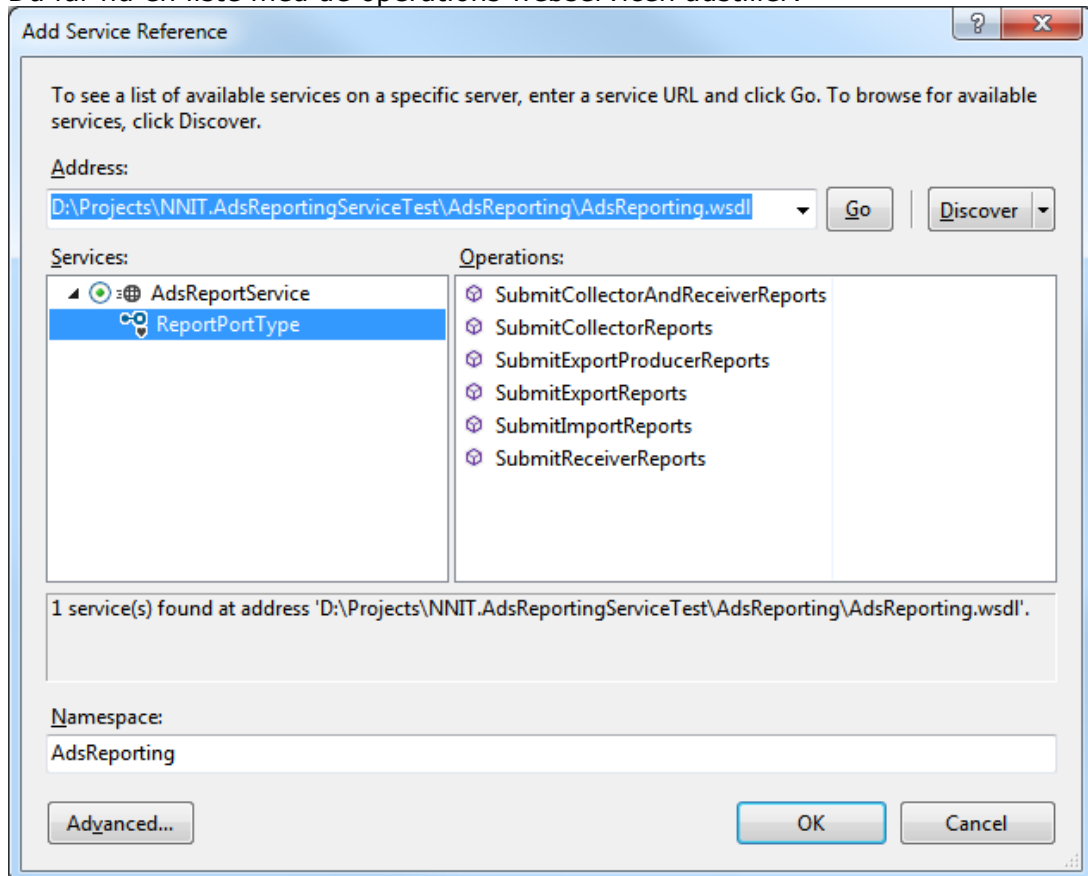


5. Indtast stien til WSDL'en som du pakkede ud fra zip-filen og tryk "Go".

6.

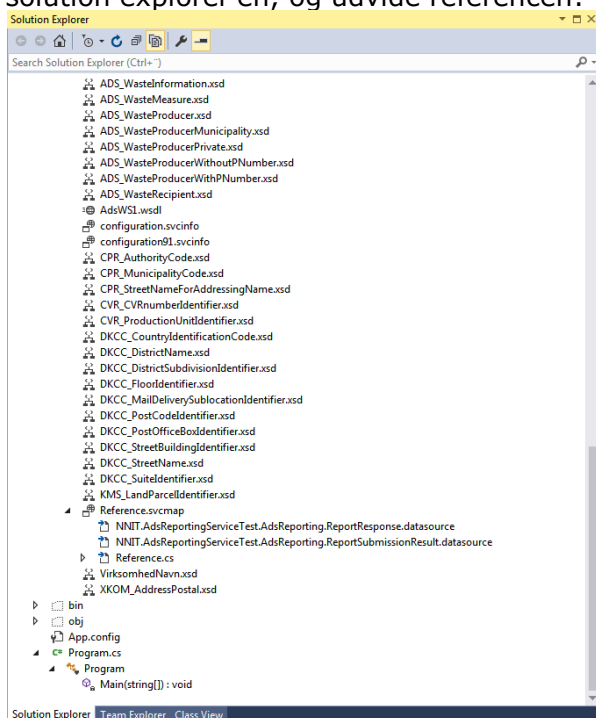


7. Du får nu en liste med de operations webservicen udstiller:



8. Navngiv referencen og tryk OK.

9. Verificer at reference.cs er blevet oprettet, og der er autogenereret klasser, samt at WSDL og skema filer er hentet fra webservicen, ved at klikke på "Show all files" i solution explorer'en, og udvide referencen:



10. Du er nu klar til at oprette instanser af de autogeneratede klasser, der kan anvendes i kald til webservicen. Eksempelvis kan man anvende instanser af `ReportPortTypeClient` og `CollectorAndReceiverInputType` til at indsende en "Indsamler og modtager" indberetning.
11. I dette eksempel oprettes et instans af klienten, og der angives et certifikat som antages at være installeret i Local Computer/Personal, og kan identificeres på thumbprint. Der indsendes en indberetning for året 2016, og for produktionsnummeret 1003405438.

```
using NNIT.AdsReportingServiceTest.AdsReporting;
using System;
using System.Security.Cryptography.X509Certificates;

namespace NNIT.AdsReportingServiceTest
{
    class Program
    {
        static void Main(string[] args)
        {
            {
                //Create instance of web service client
                var client = new ReportPortTypeClient("AdsReportPort");

                //Set certificate to use
                client.ClientCredentials.ClientCertificate.SetCertificate(
                    StoreLocation.LocalMachine,
                    StoreName.My,
                    X509FindType.FindByThumbprint,
                    "da 47 51 19 d8 77 1b d7 90 17 3e 75 41 d0 0b f7 81 f7 9a 51"
                );
                client.ClientCredentials.SupportInteractive = true;
                //create instance of request type, for submitting a "Collector and reciever" report
                var request = new CollectorAndReceiverInputType();
                CollectorAndReceiverReportType report = GenerateReport();
                request.CollectorAndReceiverReport = new[] { report };
                request.ProductionUnitIdentifier = "1003405438";
                request.ReportYear = "2016";

                try
                {
                    //Submit report to service
                    ReportSubmissionResult response =
                        client.SubmitCollectorAndReceiverReports(request);

                    //Output result to console
                    if (response != null)
                    {
                        Console.WriteLine(response.serviceResponse);
                    }
                    else
                    {
                        Console.WriteLine("Service returned empty response");
                    }
                }
                catch (System.ServiceModel.FaultException exception)
                {
                    Console.WriteLine(
                        String.Format("The service returned a fault message. Code: {0}. Message: {1}",
                            exception.Code.Name, exception.Message));
                }
                catch (Exception exception)
                {
                    Console.WriteLine(exception.Message);
                }
            }
        }
    }
}
```

```
private static CollectorAndReceiverReportType GenerateReport()
{
    CollectorAndReceiverReportType report = new CollectorAndReceiverReportType();
    report.WasteProducer = new WasteProducerType();
    report.WasteProducer.Item = new WasteProducerMunicipalityType
    {
        MunicipalityCode = "0101"
    };
    report.EAKcode = "01.01.01";
    report.TreatmentOfWaste = new TreatmentOfWasteType
    {
        FinalProcessingIndicator = true,
        WasteActivityCode = "01",
        ItemElementName = ItemChoiceType.DisposalCode,
        Item = "D1"
    };
    report.WasteInformation = new WasteInformationType
    {
        WasteCode = "H01"
    };
    report.WasteAmount = new WasteAmountType
    {
        WasteMeasure = "1,8",
        MeasurementTypeCode = 1
    };
    return report;
}
}
```

3.2 Opsætning i App.config

I App.config filen skal der lige ændres en smule så autentifikationen benytter det angivne certifikat.

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.5.2" />
  </startup>
  <system.serviceModel>
    <bindings>
      <basicHttpBinding>
        <binding name="AdsReportingBinding">
          <security mode="Transport" />
        </binding>
      </basicHttpBinding>
    </bindings>
    <client>
      <endpoint address="https://udd.ws.ads.mst.dk/ReportSubmitService.svc"
        binding="basicHttpBinding" bindingConfiguration="AdsReportingBinding"
        contract="AdsReporting.ReportPortType" name="AdsReportPort" />
    </client>
  </system.serviceModel>
</configuration>
```

Skal ændres til:

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.5.2" />
  </startup>
  <system.serviceModel>
    <bindings>
      <basicHttpBinding>
        <binding name="AdsReportingBinding">
          <security mode="Transport">
            <transport clientCredentialType="Certificate" />
          </security>
        </binding>
      </basicHttpBinding>
    </bindings>
    <client>
      <endpoint address="https://udd.ws.ads.mst.dk/ReportSubmitService.svc"
        binding="basicHttpBinding" bindingConfiguration="AdsReportingBinding"
        contract="AdsReporting.ReportPortType" name="AdsReportPort" />
    </client>
  </system.serviceModel>
</configuration>
```

Bemærk at configurationen peger på uddannelsesmiljøet

address="<https://udd.ws.ads.mst.dk/ReportSubmitService.svc>", dette skal ændrestil

address="<https://www.ws.ads.mst.dk/ReportSubmitService.svc>" i produktionsmiljøet.

3.3 Opsætning og brug af service channel

Nedenstående kode, et eksempel på anvendelse af webservicen, ved at definere et endpoint, binding og en request SOAP besked, der anvendes ved at oprette en såkaldt channel til webservicen, og kalde den derigennem. Altså et eksempel der kalder servicen, uden at anvende autogenereret kode.

Der angives et certifikat som antages at være installeret i Current user/Personal, og kan identificeres på thumbprint.

```
const string url = "https://udd.ws.ads.mst.dk/ReportSubmitService.svc";

//Create binding for web service, with certificate authentication
var binding = new BasicHttpBinding();
binding.Security.Mode = BasicHttpSecurityMode.Transport;
binding.Security.Transport.ClientCredentialType = HttpClientCredentialType.Certificate;

//Create end point at web service url
var address = new EndpointAddress(url);

//Create new channel factory, using interface which defines the service contract
var cf = new ChannelFactory<IReportSubmitService>(binding, address);

//Set credentials on channel = certificate expected to be found in Current user/Personal"
cf.Credentials.ClientCertificate.SetCertificate(StoreLocation.CurrentUser, StoreName.My,
X509FindType.FindByThumbprint,
"da 47 51 19 d8 77 1b d7 90 17 3e 75 41 d0 0b f7 81 f7 9a 51");
cf.Credentials.SupportInteractive = true;

//Open channel factory and create service channel
cf.Open();
IReportSubmitService service = cf.CreateChannel();

//Create XML reader for, which reads content of SOAP body to send in request
var xmlTextReader = new XmlTextReader(@"AdsTestRequestSoapMessageBody.xml");

//Create request message
var request = Message.CreateMessage(MessageVersion.Soap11, "Message_From_Ads_Test_Program",
xmlTextReader);

//Call web service, using constructed message
Message response = service.ProcessMessage(request);

//Close the channel factory
if (cf.State == CommunicationState.Opened)
    cf.Close();

//Is response empty?
if (response.IsEmpty)
{
    Console.WriteLine("Service returned empty response");
    return;
}

//Did the service return a fault?
if (response.IsFault)
{
    Console.WriteLine("The service returned a fault message");
}

//Deserialize the body contents of the response message
var ms = new MemoryStream();
using (var memWriter = XmlDictionaryWriter.CreateTextWriter(ms, Encoding.UTF8, false))
{
    response.WriteBodyContents(memWriter);
    memWriter.Flush();
}
```

```
//Write the message to console
Console.WriteLine(Encoding.UTF8.GetString(ms.ToArray()));
```

For at oprette en channel er det nødvendigt at definere et interface, der definerer service kontrakten. Her kan man udnytte at ADS webservice er en "catch all", der fanger alle typer kald, og behandler dem afhængigt af kaldets body. Dermed kan man nøjes med at definere en operation:

```
using System.ServiceModel;
using System.ServiceModel.Channels;

namespace NNIT.AdsReportingServiceTest
{
    [ServiceContract]
    public interface IReportSubmitService
    {
        [OperationContract(IsOneWay = false, Action = "*", ReplyAction = "*")]
        Message ProcessMessage(Message message);
    }
}
```

I eksemplet læser vi selve indholdet af den request SAOP besked der sendes, fra en fil. Denne fil skal således indeholde XML der matcher de skemaer der er defineret i webservicebeskrivelsen og beskrevet i [1]. Vil man f.eks. sende en request, der sender en "indsamler og modtager" indberetning, kunne indholdet se således ud:

```
<CollectorAndReceiverInput xmlns="http://rep.oio.dk/mst.dk/xml/schemas/2010/04/01/">
  <ReportYear>2010</ReportYear>
  <ProductionUnitIdentifier
xmlns="http://rep.oio.dk/cvr.dk/xml/schemas/2005/03/22/">1004800695</ProductionUnitIdentifier>
  <CollectorAndReceiverReport>
    <WasteProducer>
      <WasteProducerMunicipality>
        <MunicipalityCode
xmlns="http://rep.oio.dk/cpr.dk/xml/schemas/core/2005/03/18/">0151</MunicipalityCode>
      </WasteProducerMunicipality>
    </WasteProducer>
    <WasteAmount>
      <WasteMeasure>12,121</WasteMeasure>
      <MeasurementTypeCode>1</MeasurementTypeCode>
      <MeasurementText>Container</MeasurementText>
    </WasteAmount>
    <EAKcode>12.12.12</EAKcode>
    <WasteInformation>
      <WasteCode>E21</WasteCode>
    </WasteInformation>
    <TreatmentOfWaste>
      <WasteActivityCode>55</WasteActivityCode>
      <FinalProcessingIndicator>true</FinalProcessingIndicator>
    </TreatmentOfWaste>
    <DisposalCode>D6</DisposalCode>
  </CollectorAndReceiverReport>
</CollectorAndReceiverInput>
```

4 Change log

Date	Version	Beskrivelse	Initialer
2010-09-27	1.0	Første version	LNJa
2010-09-27	1.1	Tilføjet afsnit der viser brug af service channel	LNJa
2010-10-20	1.2	Mindre rettelser	LNJa
2011-02-04	2.0	Revideret version. Indeholdende mindre rettelser og kontaktinformation.	LNJa
2015-11-17	2.1	Affaldsdatasystemets adresse er opdateret Henvisninger til TEST er delvis slettet.	arguz
2015-12-02	2.1	Henvisninger til test er slettet men ikke i figurer	Jeje
2016-04-05	2.2	Afsnit 3 omskrevet, brug af VS2015 i stedet for VS2008. Import af service fra wsdl-fil i stedet for URL der ikke virker i Visual Studio (ikke muligt at anvende certifikat).	KnuC
2019-12-02	2.3	Ændring af adresse	JEJE