Component Enabler for .NET

Using the ASP.NET Web Services Generator

Overview

- Overview of Web Services
- ASP.NET Web Services
- Web Services and AB Suite
- ASP.NET Web Services Generator
- Rules for generating ASP.NET Web Services
- ASP.NET Web Services Generator setup
- Generating ASP.NET Web Services
- Editing the Web.config file
- Testing ASP.NET Web Services

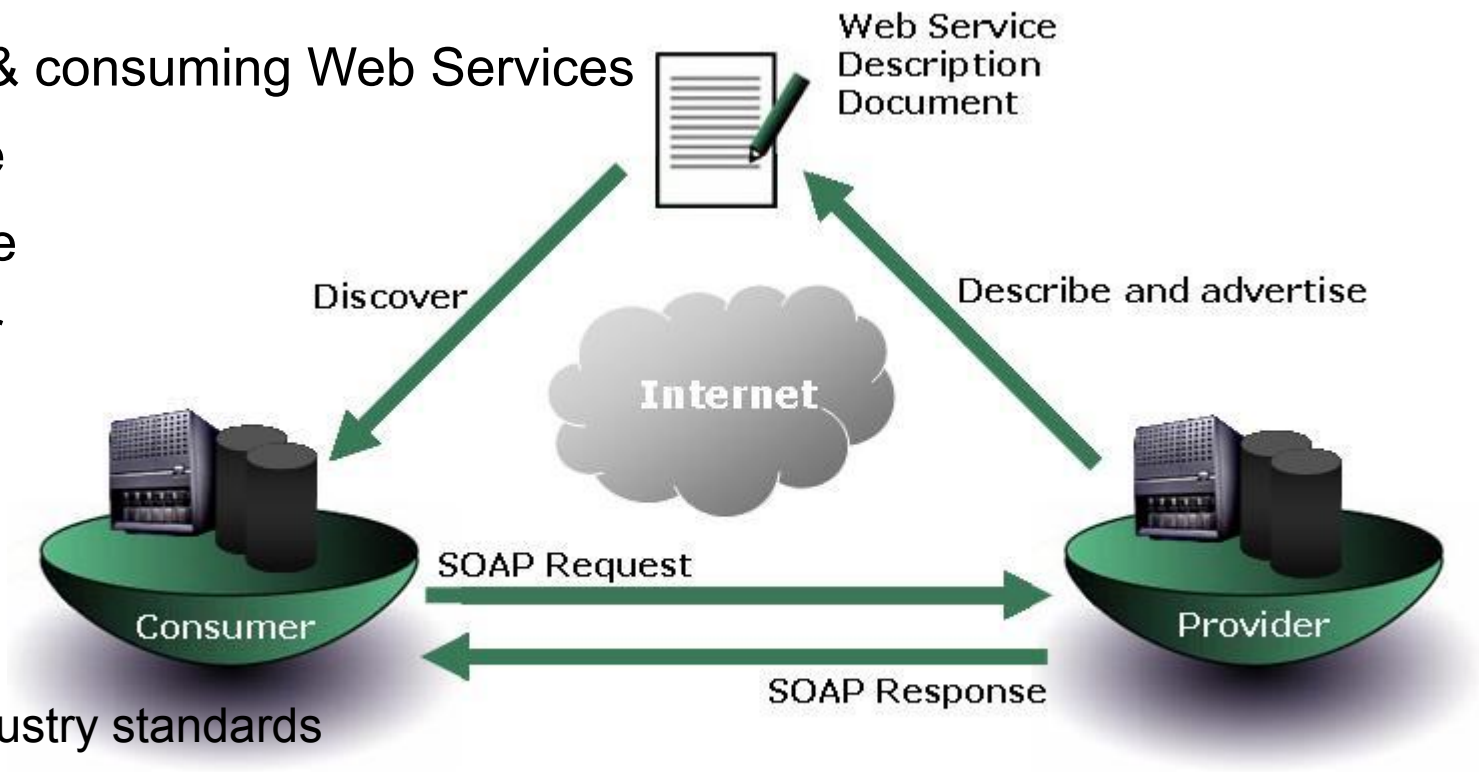
What are Web Services?

- A Web Service is a module, which can be invoked remotely using the internet/intranet infrastructure.
- Built on top of widely accepted internet standards such as TCP/IP, HTTP, Java, HTML, and XML
- Use standard technologies such as SOAP and XML for messaging, and WSDL (Web Service Description Language) for publishing
- ASP.NET Web Services are Web Services built on top of the .NET framework

Concept of Web Services

Providing & consuming Web Services

- Describe
- Advertise
- Discover
- Invoke



Based on industry standards

- XML
- WSDL (Web Service Description Language)
- SOAP (Simple Object Access Protocol)

WSDL and SOAP standards

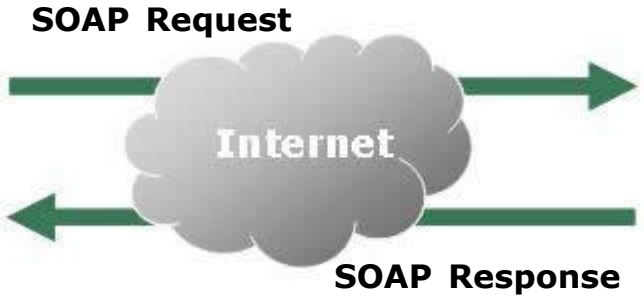
- Web Service Description Language (WSDL)
  - Where is the Web Service located?
  - What are the methods available?
  - What are the input parameters and type?
  - What is being returned?

http://www.w3.org/TR/wsdl

Web Service Description Document

- Simple Object Access Protocol (SOAP)
  - Defines the message format
  - HTTP and XML based
  - Defines fault message format

http://www.w3.org/TR/SOAP



WSDL Example – 1/2

```
<?xml version= '1.0' encoding='UTF-8'?>

<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
    xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
    xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
    xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:tns="http://www.MyCompany.com/"
    xmlns:s=http://www.w3.org/2001/XMLSchema>

 <wsdl:types>
    <s:schema elementFormDefault="qualified" targetNamespace="http://www.MyCompany.com/">
    <s:element name="CUSTOMER_COMPONENT_Delete">
     <s:complexType>
     <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="CUSTOMER_COMPONENT_DeleteRequest"
    type="tns:CUSTOMER_COMPONENT_Delete_Input" />
      </s:sequence>
      </s:complexType>
      </s:element>
    ------
    ------
    </s:schema>
    </wsdl:types>

  <wsdl:message name="CUSTOMER_COMPONENT_DeleteSoapIn">
  <wsdl:part name="parameters" element="tns:CUSTOMER_COMPONENT_Delete" />
  </wsdl:message>

<wsdl:message name="CUSTOMER_COMPONENT_DeleteSoapOut">
  <wsdl:part name="parameters" element="tns:CUSTOMER_COMPONENT_DeleteResponse" />
  </wsdl:message>
```

Input parameter type

Return parameter and types

WSDL Example – 2/2

```
<wsdl:portType name="wservicesSoap">
      <wsdl:operation name="CUSTOMER_COMPONENT_Delete">
      <wsdl:input message="tns:CUSTOMER_COMPONENT_DeleteSoapIn" />
      <wsdl:output message="tns:CUSTOMER_COMPONENT_DeleteSoapOut" />
  </wsdl:operation>

 <wsdl:binding name="wservicesSoap" type="tns:wservicesSoap">
   <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
      <wsdl:operation name="CUSTOMER_COMPONENT_Delete">
        <soap:operation
      soapAction="http://www.MyCompany.com/CUSTOMER_COMPONENT_Delete"
      style="document" />
        <wsdl:input> <soap:body use="literal" /> </wsdl:input>
        <wsdl:output> <soap:body use="literal" /> </wsdl:output>
      </wsdl:operation>
 </wsdl:binding>

 <wsdl:service name="wservices">
 <wsdl:port name="wservicesSoap" binding="tns:wservicesSoap">
  <soap:address location="http://localhost/SAMPLE_WSERVICES_WS/WSERVICES.asmx"
      />
  </wsdl:port>
</wsdl:service>

</definitions>
```

Available Methods

Transport Types and Message Format

Service location

SOAP Request Message

```
<?xml version="1.0" encoding="utf-8"?>
 <soap:Envelope
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>
  <CUSTOMER_COMPONENT_Delete xmlns="http://www.MyCompany.com/">
    <CUSTOMER_COMPONENT_DeleteRequest>
      <Customer Number>string</Customer Number>
    </CUSTOMER_COMPONENT_DeleteRequest>
  </CUSTOMER_COMPONENT_Delete>
</soap:Body>
</soap:Envelope>
```

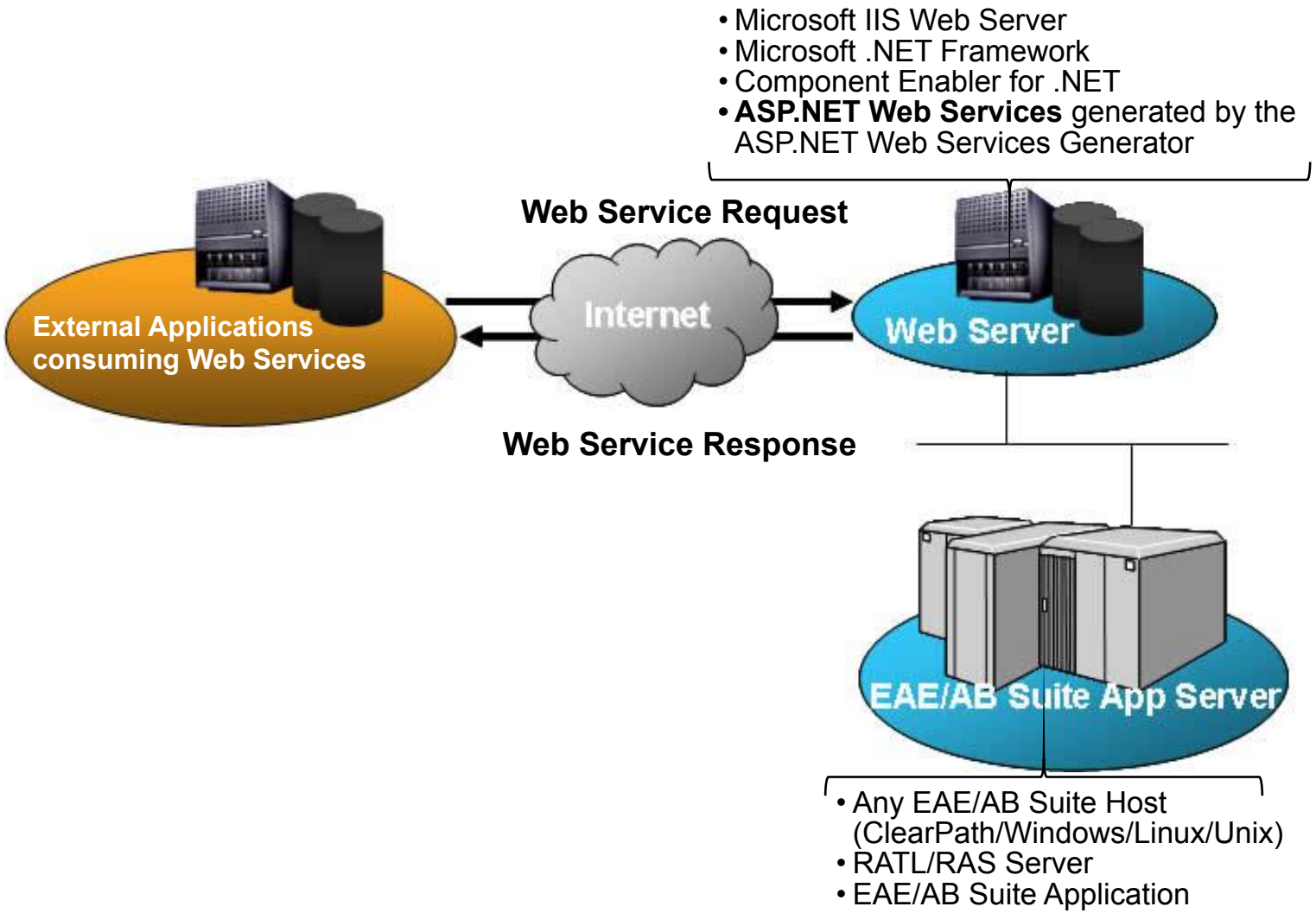
SOAP Response Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema"
      xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <CUSTOMER_COMPONENT_DeleteResponse
      xmlns="http://www.MyCompnay.com/">
      <CUSTOMER_COMPONENT_DeleteResult>
        <StatusMessages>
          <StatusMessage>string</StatusMessage>
          <StatusMessage>string</StatusMessage>
        </StatusMessages>
      </CUSTOMER_COMPONENT_DeleteResult>
    </CUSTOMER_COMPONENT_DeleteResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP Fault Message

```
<?xml version="1.0" encoding="UTF-8" ?>
    <soap:Envelope
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema"
   xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
      <soap:Body>
        <soap:Fault>
          <faultcode>SOAP:Client</faultcode>
          <faultstring>Customer number not found</faultstring>
          <detail>Enter a valid customer number</detail>
        </soap:Fault>
      </soap:Body>
    </soap:Envelope>
```

ASP.NET Web Services Environment



ASP.NET Web Services Generator

- Generates a Web Services interface which is based on the Web Services infrastructure provided by Microsoft .NET technology
- Takes advantage of .NET Framework and ASP.NET support for Web Methods
- Generates Web Service interface for selected ispecs
- Allows security using WSE for .NET

Rules for generating Web Services – 1/2

Ispecs with persistent attributes

- One Web Service method generated for each MAINT function
- Options in the GeneratorConfig.xml file specify which MAINT functions to generate

Ispecs with non-persistent attributes

- One Web Service method generated for each ispec

Web Service input fields

- Field usage (*Direction* and *IsPersistent* settings on an attribute) settings on an attribute determine whether it is generated as a web service method

Input ispec must be the same as Output Ispec

- Recall of another ispec will return an error at run time

Web Service output fields

- Options in GeneratorConfig.xml file specify how to interpret ispec field usage when generating output parameters from a Web Service method

Rules for generating Web Services – 2/2

System specific fields

- Status-line and multiple error lines are generated as output from a Web Service method
- Other system-specific fields such as MAINT and ACTMTH fields will not be returned

Presentation attributes

- Attributes with presentation are generated as input according to the rules for input fields

Generating meaningful names

- Ispec Alias or ispec Description used as Web Service method name
- Data Name or Data Caption used as XML tag names
- Options in the GeneratorConfig.xml file specify whether short or long names generated

Stateless systems

- Web Services assume a stateless environment
- Every call from an external application is considered a new transaction

ASP.NET Web Services Generator Setup

- Install ASP.NET Web Services generator
- Initialize Bundle View using the InitializeBundleView wizard. The script:
  - Creates the directory structure for a bundle view
  - Copies all necessary infrastructure files into the views directory
  - Registers an IIS virtual directory associated with a views directory

Editing Generatorconfig.xml

An example of Generatorconfig.xml

```
<?xml version="1.0" encoding="utf-8" ?>
- <configuration>
  <maint INQ="Yes" REC="No" DEL="Yes" NEX="No" BAC="No" CHG="YES" ADD="Yes"
      FIR="No" LAS="No" PUR="No" />
  <output usageInput="Yes" usageIO="Yes" usageInquiry="Yes" />
  <listItemsOutput dynamicList="Yes" staticList="Yes" />
  <naming ispecDescription="Yes" dataDisplay="Yes" />
  <deployment namespace="www.unisys.com" virtualDirectory="SAMPLE_bundle_WS"
      />
  </configuration>
```

Set the following attributes in GeneratorConfig.xml

- Maint functions : INQ, REC, DEL, NEX, BAC, CHG, ADD, FIR, LAS, PUR
- Output field usage: usageInput, usageIO, usageInquiry
- List items output: dynamicList, staticList
- Naming: ispecDescription, dataDisplay
- Deployment: namespace, virtualDirectory

Generating an ASP.NET Web Services bundle

Sample bundle Configuration Properties

Category	Property	Value
Build Target Filter	Deploy Component Enabler User Interface	True
General	Deployable	True
	Start Deployment With	Generate
	Stop Deployment After	Generate
Component Enabler User Interface	User Defined View Generator	GenerateWSdotNET.dll
	Ispec Models Source Language	C#
Installation	Package Name	bundle (name for the bundle)
	Deployment Host	localhost

Configuring ASP.NET Web Services

- This is a standard .NET xml formatted configuration file.
- Similar to ASP.NET WebForms
- Values stored in <appSettings> contains configuration parameters specific to the host runtime system.
- Edit the Web.config file to customize it for your application.

```
- <appSettings>
    <add key="LINCEnvironmentProgId" />
    <add key="ApplicationName" value="SAMPLE" />
    <add key="BundleName" value="bundle" />
    <add key="DisplayName" value="Display Name" />
    <add key="PackagePrefix" value="com.unisys" />
    <add key="HostURI" value="x-ratl:localhost:2449" />
    <add key="HostViewName" value="SAMPLE" />
    <add key="ConnectionMode" value="0" />
    <add key="IspecModelRootDirectory"
      value="C:\NGEN_CE\classes" />
    <add key="HostLogin" value="AppUser" />
    <add key="HostPassword" value="ngen_User1" />
    <add key="HostDomain" value="." />
    <add key="LoggingEnabled" value="true" />
    <add key="LogFile" value="C:\Temp\CETraces.log" />
    <add key="LogLevel" value="7" />
    <add key="ReturnTransactionErrorAsSoapFault"
      value="true" />
    <add key="ObjectPoolingEnabled" value="false" />
    <add key="ObjectPoolingAssembly" />
    <add key="ObjectPoolingType" />
    <add key="EnableSoapTrace" value="false" />
    <add key="TraceFile"
      value="C:\Temp\CEdotNetWsLog.txt" />
    <add key="EnableSchemaValidation" value="false" />
  </appSettings>
```

Deploying an ASP.NET Web Service

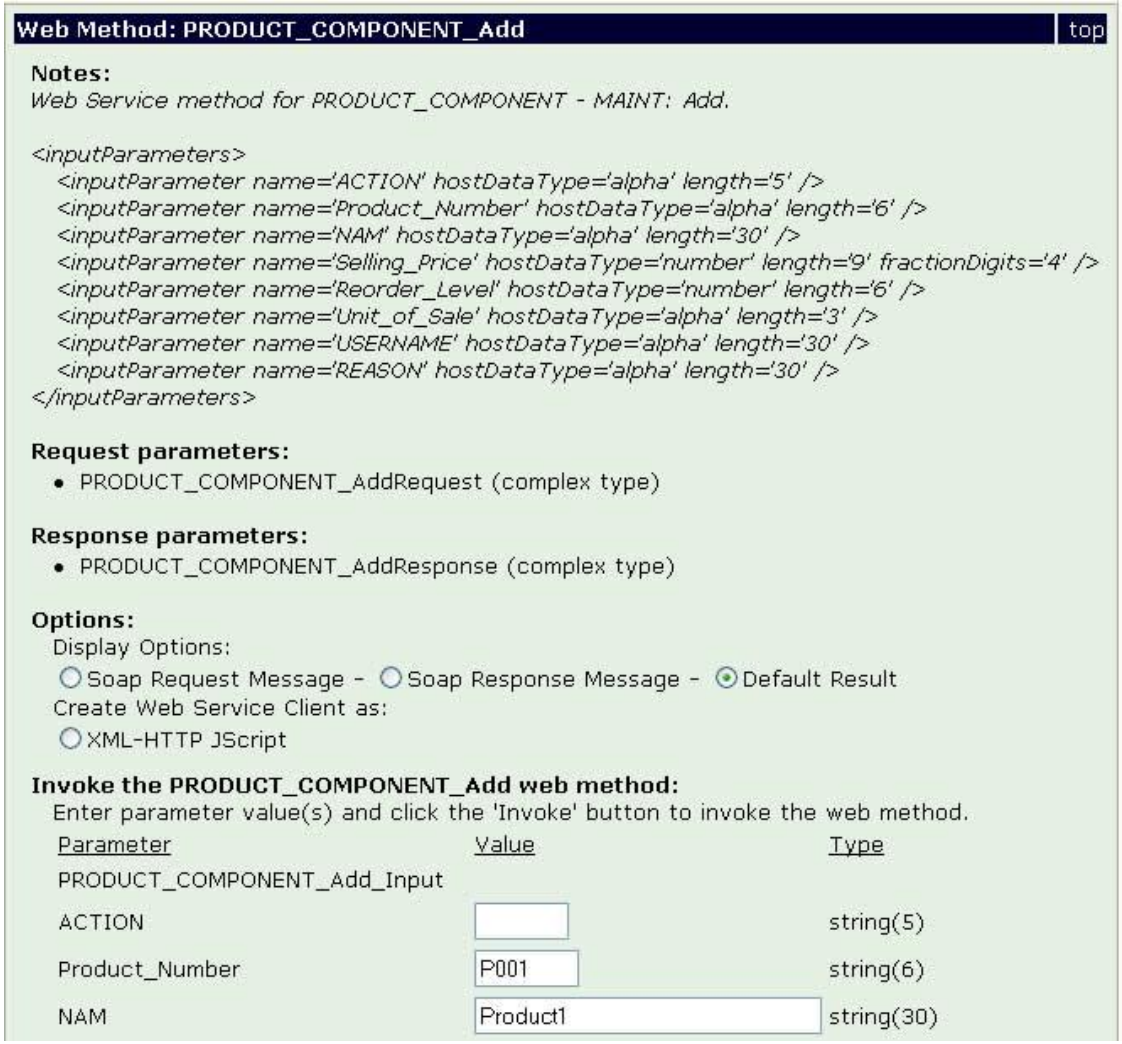
- Compile the generated Web Services by running the CompileWebService.bat file, which is located in the views directory of the bundle.
- Copy the compiled files from the views folder to the Web server
- Install Runtime for .NET Framework on the Web server

Testing ASP.NET Web Services – 1/2

- Can use the Discovery tool to test your Web Services. This tool is available under the <CE installation directory>\Web Services .NET Generator\utilities.
- To set up the Discovery tool refer to the ReadmeDiscover.txt file.
- The Web Service Discovery tool is unsupported software.
- To test your Web Services, access the Discovery tool, enter the URL of the generated Web Services in the Discover field and click **Discover**.

Testing ASP.NET Web Services – 2/2

- The WSDL file gets created on the fly.
- The Discovery Tool simulates a Web Service client by:
  - Reading and analyzing the content of the WSDL file
  - Dynamically building a browser form for each method
  - Building a SOAP request method and calling the Web Service method
  - Receiving the SOAP response method and displaying the results



Summary

- A Web Service is a module that can be invoked remotely using the internet/intranet infrastructure.
- ASP.NET Web Services are applications based on the Microsoft ASP.NET infrastructure.
- In AB Suite, ASP.NET Web Services are generated using the ASP.NET Generator.
- ASP.NET Web Services generated from AB Suite can be tested using the Discovery tool.
- The generated ASP.Net Web Services can be invoked by any external client that uses standard protocols such as SOAP and XML.