

Examples

This chapter covers the following topics:

- Example 1: Existing Natural Client that Connects to a Web Service
- Example 2: Publish an EntireX RPC Server for Web Clients

Example 1: Existing Natural Client that Connects to a Web Service

Natural Program

```
* CALC      : CLIENT                                     *
* ----- *
DEFINE DATA
LOCAL
  01 #OPERATOR          (A1)
  01 #OPERAND1          (I4)
  01 #OPERAND2          (I4)
  01 #RESULT            (I4)
*
  01 #MSG                (A60)
END-DEFINE
* ----- *
ASSIGN #OPERAND1 = 12345
ASSIGN #OPERATOR = '+'
ASSIGN #OPERAND2 = 67890
ASSIGN #RESULT   = 0
*
CALLNAT 'CALC' #OPERATOR (AD=O)
              #OPERAND1 (AD=O)
              #OPERAND2 (AD=O)
              #RESULT   (AD=A)
*
COMPRESS #OPERAND1 #OPERATOR #OPERAND2 '=' #RESULT
        INTO #MSG LEAVING NO SPACE
*
PRINT #MSG
* ----- *
END
```

- Use the IDL Extractor for Natural to get the IDL file. In the EntireX perspective, use **New > IDL Extractor for Natural** . In other perspectives, use **New > Other... > Software AG > IDL Extractor for Natural**.

Software AG IDL

```
Library 'Example' Is
Program 'Calc' Is
  Define Data Parameter
    1 Operator          (A1) In
```

```

1 Operand_1      (I4) In
1 Operand_2      (I4) In
1 Function_Result (I4) Out
End-Define

```

- Select the IDL file and open the context menu.
- Choose **Web Service > Generate Web Service...**, and the XMM and WSDL file will be generated. See *EntireX Web Services Wrapper*.

Example.wsdl

```

<?xml version="1.0" encoding="utf-8"?>
<definitions name="example" targetNamespace="http://namespace.softwareag.com/entirex/xml/mapping"
  xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:tns="http://namespace.softwareag.com/entirex/xml/mapping" xmlns:sn0="urn:com-softwareag-entirex-rpc:EXAMPLE">
  <types>
    <xsd:schema targetNamespace="urn:com-softwareag-entirex-rpc:EXAMPLE">
      <xsd:element name="CALC">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="Operator" type="xsd:string"/>
            <xsd:element name="Operand_1" type="xsd:int"/>
            <xsd:element name="Operand_2" type="xsd:int"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="CALCResponse">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="Function_Result" type="xsd:int"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:schema>
  </types>
  <message name="CALCSoapIn">
    <part name="parameters" element="sn0:CALC"/>
  </message>
  <message name="CALCSoapOut">
    <part name="parameters" element="sn0:CALCResponse"/>
  </message>
  <portType name="EXAMPLEPort">
    <operation name="CALC">
      <input message="tns:CALCSoapIn"/>
      <output message="tns:CALCSoapOut"/>
    </operation>
  </portType>
  <binding name="EXAMPLESOAP11Binding" type="tns:EXAMPLEPort">
    <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="CALC">
      <soap:operation soapAction="CALC"/>
      <input>
        <soap:body use="literal"/>
      </input>
      <output>
        <soap:body use="literal"/>
      </output>
    </operation>
  </binding>
  <binding name="EXAMPLESOAP12Binding" type="tns:EXAMPLEPort">
    <soap12:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="CALC">
      <soap12:operation soapAction="CALC"/>
      <input>
        <soap12:body use="literal"/>
      </input>
      <output>
        <soap12:body use="literal"/>
      </output>
    </operation>
  </binding>
  <service name="example">
    <port name="EXAMPLESOAP11Port" binding="tns:EXAMPLESOAP11Binding">
      <soap:address location="http://localhost:10010/wsstack/example"/>
    </port>
    <port name="EXAMPLESOAP12Port" binding="tns:EXAMPLESOAP12Binding">

```

```

        <soap12:address location="http://localhost:10010/wsstack/example"/>
    </port>
</service>
</definitions>

```

- Create a service skeleton with Apache Axis.

```
java -classpath org.apache.axis.wsdll.WSDL2Java --server-side --skeletonDeploy true Example.wsdll
```

- And write an implementation of this service.

```

/**
 * Service.java
 *
 * Implementation of ExamplePort
 * generated by the Apache Axis WSDL2Java emitter.
 */

package com.softwareag.namespace;

public class Service implements ExamplePort
{
    public int calc(java.lang.String operator_, int operand_1, int operand_2)
        throws java.rmi.RemoteException
    {
        int result = 0;
        if (operator_.equals("+"))
        {
            result = operand_1 + operand_2;
        }
        else if (operator_.equals("-"))
        {
            result = operand_1 - operand_2;
        }
        else if (operator_.equals("*"))
        {
            result = operand_1 * operand_2;
        }
        else if (operator_.equals("/"))
        {
            result = operand_1 / operand_2;
        }
        return result;
    }
}

```

- Build and deploy the service (see Apache Axis documentation).
- Configure the XML/SOAP RPC Server

entirex.xmlrpcserver.properties

```

# jaxp parameters
# if jaxp properties are not set in system properties
#
# xmlruntime configuration file
entirex.sdk.xml.runtime.configurationfile=.entirex.xmlrpcserver.configuration.xml

```

entirex.xmlrpcserver.configuration.xml

```
<?xml version="1.0" encoding="iso-8859-1" ?>
<EntireX
  xmlns="http://namespaces.softwareag.com/entirex/xml/runtime/configuration" version="7.1.1">
  <XmlRuntime Version="1">
    <BrokerInfo>
      <BrokerId>localhost:1971</BrokerId>
      <ServerAddress>RPC/SRV1/CALLNAT</ServerAddress>
    </BrokerInfo>
    <TargetServer name="http://localhost:8080/axis/services/ExamplePort">
      <XmmList>
        <!--the name of XMM file-->
        <Xmm name="./Calc.xmm" />
      </XmmList>
    </TargetServer>
  </XmlRuntime>
</EntireX>
```

- start the XML/SOAP RPC Server:

```
java com.softwareag.entirex.xml.rt.XMLRPCServer
```

Example 2: Publish an EntireX RPC Server for Web Clients

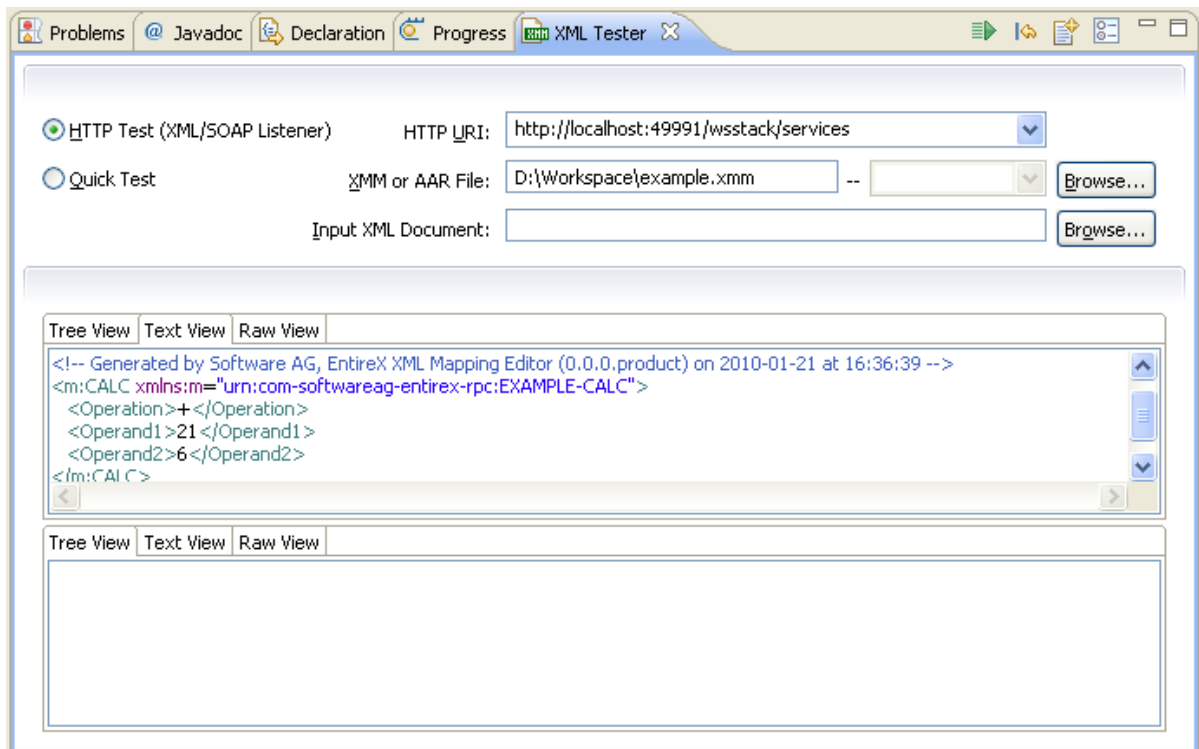
Software AG IDL File

```
library 'UserList' is
  program 'Add' is
    define data parameter
      1 Name (AV) In
    end-define

  program 'Retrieve' is
    define data parameter
      1 Name (AV/V) Out
    end-define
```

➤ To publish the EntireX RPC/SOAP server

1. Create a new IDL file *User List* (Using: **New > Software AG IDL file** or **New > Others ... > Software AG > Software AG IDL file**).
2. Select the IDL file and generate the RPC server from the context menu.
3. Select the IDL file, and from the context menu choose **Web Service > Generate Web Service...** Generate and deploy the AAR file with the Packaging Wizard.
4. Select the tab **XML Samples** in the EntireX XML Mapping Editor and save one or all sample documents.
5. Select the request document in the tree and open the XML Tester with the context menu.



6. Change the URL to the required address, for example: *http://localhost:10010/wsstack/services/example*. Choose **Send**. The response document will be displayed in the lower portion of the screen.