Java Wrapper Examples

- Delivered Java Wrapper Examples
- Running the Delivered Examples

Delivered Java Wrapper Examples

This section describes the examples for the Java Wrapper folder *examples/java_wrapper* of the Developer's Kit.

- Scope
- Software AG IDL File and Interface Object Generation Process for the Example
- Other EntireX Developer's Kit Wrapper Examples

Scope

This folder *examples/RPC/basic/example/JavaClient* contains an example of a standalone application (*MyClient.java*) that calls remote procedures CALC and SQUARE with its associated IDL file *example.idl*. An example implementation of a server interface object for the Java RPC Server is available too (*ExampleServer.java*). This server interface object runs with the generic Java RPC Server, which is part of the Java Runtime.

To run the examples from within the Eclipse IDE, import the folder *examples/RPC/basic/example/JavaClient* into a Java project, then add *entirex.jar* to the build path of this project.

This example consists of the following programs, which may be called remotely using the EntireX Broker:

- **CALC**
  Uses two operands and one operator (+-*/) to return the result. A Java client will be able to call our sample method like this:

  ```
  Example myExample = new Example(broker);
    int op1 = 1234;
    int op2 = 5678;
    int y = myExample.calc ("+", op1, op2)
  ```

- **SQUARE**
  Uses one input value to return its square.

Software AG IDL File and Interface Object Generation Process for the Example

The IDL file describes the interface. See *Software AG IDL File*. For the mapping between IDL data types and types of Java see *Mapping IDL Data Types to Java Data Types*.

This IDL file *example.idl* of the Java Wrapper example is part of the EntireX examples.

```
Library ’EXAMPLE’ Is
   Program ’CALC’ Is
      Define Data Parameter
         1 Operation       (A1)  In
         1 Operand_1       (I4)  In
         1 Operand_2       (I4)  In
         1 Function_result (I4)  Out
      End-Define

   Program ’SQUARE’ Is
      Define Data Parameter
         1 Operand         (I4)  In
         1 Result          (I4)  Out
      End-Define
```

Generating a client interface object with the Java Wrapper creates the following file in the source folder of the IDL file: *Example.java*.

Generating a server interface object with the Java Wrapper produces the following files in the source folder of the IDL file: *ExampleServer.java*, *ExampleStub.java*, *AbstractExampleServer.java*.

Other EntireX Developer’s Kit Wrapper Examples

Other Wrappers of the Developer’s Kit provide the same examples (CALC and SQUARE). The examples can be mixed, meaning any client can call any server. Natural RPC is also fully compatible with EntireX RPC. A Natural client can call any EntireX RPC server and vice versa.

For examples of other programming language bindings see:

- *Delivered Examples for the C Wrapper*
- *Delivered Examples for the COBOL Wrapper*
- Delivered Examples for Natural in subdirectory Java Wrapper, Broker RPC/Client and Broker RPC/Server.

Running the Delivered Examples

Prerequisites for Running the Examples

1. Verify that the Java classpath contains both the *entirex.jar* file, which is located in the classes directory, and an entry for the directory containing the generated classes.

2. To run the client programs, an RPC server is needed. You may use the example RPC server under CICS, UNIX and Windows. If you want to use a Natural RPC Server copy the **.nsn* files to a Natural library. You can also use the Java RPC Server under UNIX and Windows.

To run the client example

1. Generate the client interface object.

2. Compile the *MyClient.java* file.

3. Run *MyClient.class*.

4. Select one of the examples with the option button and press **Call()**.

5. If you want to use EntireX Security, uncomment the line

   ```
   //broker.setSecurity(new EntireXSecurity(), false);
   ```

To run the server example

1. Generate the RPC server as described under *Generating a Java Server Interface Object*.

2. Implement the methods in *<library name>Server.java.*

3. Run the Java RPC Server. The classpath must contain the directory of the server interface object classes. If you start the Java RPC Server in the current directory, then add "." to the classpath. If the Java RPC Server is started in a different directory, the complete path name of the Java Wrapper example has to be part of the classpath. Otherwise the Java RPC Server cannot load the server interface object classes.

4. To shut down the Java RPC Server, use the System Management Hub. (Note that this stops all RPC servers that register the same service.)