Writing Standard Call Interface Clients

This chapter describes in six steps how to write your first COBOL RPC client program.

- Step 1: Declare and Initialize the RPC Communication Area
- Step 2: Declare the Data Structures for RPC Stubs
- Step 3: Required Settings in the RPC Communication Area
- Step 4: Optional Settings in the RPC Communication Area
- Step 5: Issue the RPC Request
- Step 6: Examine the Error Code

The following steps describe how to write a COBOL client program for the client scenarios: *Micro Focus* | *Batch* | *CICS* | *IMS*. We recommend reading them first before writing your first RPC client program and following them if appropriate.

The example given here does not use function calls as described under *Using Broker Logon and Logoff*. It demonstrates an implicit broker logon (because no broker logon/logoff calls are implemented), where it is required to switch on the `AUTOLOGON` feature in the broker attribute file.

Step 1: Declare and Initialize the RPC Communication Area

The RPC communication area (see *Using the RPC Communication Area*) is your interface (API) to the *Generic RPC Services Modules*. Declare and initialize the communication area in your RPC client program as follows:

```
* Declare RPC communication area
 01 ERX-COMMUNICATION-AREA EXTERNAL.
    COPY ERXCOMM.

* Initialize RPC communication area
 INITIALIZE ERX-COMMUNICATION-AREA.
 MOVE "2000" to COMM-VERSION.
```

The example given here uses option `External Clause` to access the RPC communication area. See *Using the RPC Communication Area with a Standard Call Interface*. For further options to access the RPC communication area, see *RPC Communication Area*.

Step 2: Declare the Data Structures for RPC Stubs

For every program definition of the IDL file, the COBOL Wrapper generates a copybook with the description of the customer's interface data as a COBOL structure. For ease of use you can include these structures into your RPC client program. See *Using the Generated Copybooks*.

However, as an alternative, you can use your own customer data structures. In this case the COBOL data types and structures must match the interfaces of the generated client interface objects, otherwise unpredictable results may occur.

```
* Declare customer data to generated RPC Stubs
 01 CALC-AREA.
    10 PARAMETER.
       15 OPERATOR                   PIC X.
       15 OPERAND1                   PIC S9(9) BINARY.
       15 OPERAND2                   PIC S9(9) BINARY.
       15 RESULT                     PIC S9(9) BINARY.
```

Step 3: Required Settings in the RPC Communication Area

The following settings to the RPC communication area are required as a minimum to use the COBOL Wrapper. These settings have to be applied in your RPC client program. It is not possible to generate any defaults into the client interface objects.

```
* assign the broker to talk with ...
 MOVE "localhost:1971" to COMM-ETB-BROKER-ID.
* assign the server to talk with ...
 MOVE "RPC"            to COMM-ETB-SERVER-CLASS.
 MOVE "SRV1"           to COMM-ETB-SERVER-NAME.
 MOVE "CALLNAT"        to COMM-ETB-SERVICE-NAME.
* assign the user id to the broker ...
 MOVE "ERXUSER"        to COMM-USERID.
 MOVE "PASSWORD"       to COMM-PASSWORD.
```

Step 4: Optional Settings in the RPC Communication Area

Here you specify optional settings to the RPC communication area used by the COBOL Wrapper, for example:

```
 MOVE "EXAMPLE"        to COMM-LIBRARY.
 MOVE "00000300"       to COMM-ETB-WAIT.
```

For implicit broker logon, if required in your environment, the client password can be given here. It is provided then through the client interface objects, see also *Using Broker Logon and Logoff*.

Step 5: Issue the RPC Request

Issue the RPC request with a standard COBOL program call:

```
 CALL "CALC" USING  OPERATOR OPERAND1 OPERAND2 RESULT.
```

Step 6: Examine the Error Code

When the RPC reply is received, check that the call was successful:

```
 IF COMM-RETURN-CODE IS = ZERO
    Perform success-handling
 ELSE
    Perform error-handling
 END-IF.
```

The field `COMM-RETURN-CODE` in the RPC communication area contains the error provided by the COBOL Wrapper. For the error messages returned, see *Error Messages and Codes*.