Installing the EntireX RPC Server under IBM i

The EntireX RPC Server under IBM i enables you to call programs as servers, using ILE (Integrated Language Environment).

Note:
The EntireX RPC server installation also includes sample programs that demonstrate how to build and use an RPC client environment on IBM i. For details, see *Installation Verification* and *Using the COBOL Wrapper*.

This chapter describes how to install the RPC server under IBM i.

It covers the following topics:

- Features Currently not Supported
- Installation Steps
- Installation Verification

See *Platform Coverage* for full platform information.

Prerequisites for installing the RPC server are described centrally. See *IBM i Prerequisites*.

The implementation under IBM i is based on the UNIX code, therefore, use the UNIX parameters for the IBM i environment, see *Setting up Broker Instances*. However, some features provided under UNIX are not supported under IBM i, see *Features Currently not Supported*.

Features Currently not Supported

Secure Sockets Layer (SSL) is currently not supported.

Installation Steps

Note:
vrsp stands for the current version, release, service pack, and optionally a patch level.

Installation comprises the following steps:

- Step 1: Check the Product Library
- Step 2: Copy the Installation Kit to Disk
- Step 3: Verify the Contents of the *SAVF File
- Step 4: Restore the *SAVF File

Step 1: Check the Product Library

A successful Broker ACI installation on your IBM i machine is a prerequisite to install the RPC server. Software AG recommends you to keep the RPC server environment in the library EXX, in which your Broker ACI was installed.

If this library does not yet exist, create it with the command `CRTLIB EXX`.

Step 2: Copy the Installation Kit to Disk

The product is delivered in a data set with the name *../OS400/EXPvrsp* on your EntireX installation DVD.

To copy the installation file to your IBM i disk

1. Use the command `CRTSAVF` to create an empty IBM i `*SAVF` file, named EXP *vrsp*, on your IBM i machine in a library of your choice.
2. Use FTP to transfer the unzipped PC file EXP *vrsp* to the corresponding IBM i save file using the FTP option "binary".

Step 3: Verify the Contents of the *SAVF File

To verify the contents of the *SAVF file EXP*vrsp*

- Use the IBM i command DSPSAVF. The command should display the following objects:

Object	Type	Description
XSERVER	*PGM	RPC server program.
STR_RPCSRV	*PGM	Sample RPC server starting procedure.
EXPCRTLOG	*PGM	Sample procedure to create the physical log file.
EXPRUNTIME	*SRVPGM	RPC server runtime module.
EXPMSG	*MSG	EntireX Messages.
H_EXP	*FILE (PF-SRC)	Header include files for sample C programs. Rename to H before use.
QCLSRC	*FILE (PF-SRC)	Sources of build procedures . **Member** — **Description** SERVER_CFG — RPC server configuration to be used by the server startup procedure. STR_RPCSRV — Sample procedure to submit/start an RPC server. EXPCRTLOG — Procedure to create the physical log file.
EXAMPLE	*SAVF	Save file containing RPC server and RPC client examples.

Step 4: Restore the *SAVF File

To restore the *SAVF file

- Use the command RSTLIB:

```
RSTLIB SAVLIB(EXPvrsp) DEV(*SAVF) SAVF(yoursavlib/EXPvrsp) RSTLIB(EXX)
```

where *yoursavlib* denotes the library into which you transferred the save file during *Step 2*.

In addition to the Broker ACI installation, the product library EXX should now contain the objects listed in *Step 3*.

Software AG recommends you not having user objects in this library. Copy modified samples to the user libraries. Source samples mentioned here may change, therefore check all associated program objects for the latest version.

The EntireX RPC server for IBM i is now installed and ready to use.

Installation Verification

The verification is based on the EntireX RPC calculation example CALC in the IDL file *example.idl*, which is common for all platforms. The IBM i library EXAMPLE provides programs that allow you to test a COBOL RPC client and to run the RPC server with a COBOL or a C application service.

The verification comprises the following steps:

- Step 1: Restore the EXAMPLE Library
- Step 2: Verify the RPC COBOL Client
- Step 3: Verify the RPC Server using COBOL
- Step 4: Verify the RPC Server using C

Note:
RPG source examples are currently not provided for the installation verification.

Prerequisite for all verifications:

The service program EXA (type *SRVPGM, the Broker ACI/stub) must be available in your library list. You can accomplish this task by adding EXX, where the Broker ACI and the RPC server are usually installed, to your library list.

Step 1: Restore the EXAMPLE Library

Before you can begin the verification, you must restore the IBM i library EXAMPLE from the save file EXAMPLE that you downloaded to library EXX during *Step 3*.

To restore the *SAVF example file, use the command RSTLIB:

```
RSTLIB SAVLIB(EXAMPLE) DEV(*SAVF) SAVF(EXX/EXAMPLE) RSTLIB(EXAMPLE)
```

The newly created library EXAMPLE should contain the following objects:

Object	Type	Used by example	Description
CALC	*PGM	COBOL server, see *Step 3: Verify the RPC Server using COBOL*	COBOL server calculation engine (created with BIND_CALC).
CALC_RPG	*PGM	RPC server	RPG Server calculation engine (created with BIND_RCALC).
CALCCLIENT	*PGM	COBOL client, see *Step 2: Verify the RPC COBOL Client*	COBOL client calculation dialog (created with BINDCCALC). Contains the modules CCALCMAIN, CCALCMENU, CCALC and RPCSRVI.
CRT_CBLMOD	*PGM	all COBOL	Procedure to compile/create COBOL modules.
CRT_C_SRV	*PGM	all C	Procedure to create modules and service programs of DEXAMPLE and EXAMPLE for the C Server.
BIND_CALC	*PGM	COBOL server, see *Step 3: Verify the RPC Server using COBOL*	Procedure to compile and bind the COBOL server.
BIND_CCALC	*PGM	COBOL client, see *Step 2: Verify the RPC COBOL Client*	Procedure to compile and bind the COBOL client.
BIND_RCALC	*PGM	RPG Server	Procedure to compile and bind the RPG Server.
STR_RPCSRV	*PGM	all Server	Procedure to start the EntireX RPC server.
X_DEXAMPLE	*SRVPGM	C Server, see COBOL client, see *Step 4: Verify the RPC Server using C*	C Server generated stub service. To use it, rename to DEXAMPLE.
X_EXAMPLE	*SRVPGM	C Server, see COBOL client, see *Step 4: Verify the RPC Server using C*	C Server implementation file. This service contains the application logic of the programs CALC, HELLO and POWER. To use it, rename to EXAMPLE.

Object	Type	Used by example	Description
CCALCMENU	*FILE (DSPF)	COBOL client, see *Step 2: Verify the RPC COBOL Client*	COBOL client dialog screen.
CALC	*MODULE	COBOL client, see *Step 2: Verify the RPC COBOL Client*	COBOL server calculation engine.
QCBLLESRC	*FILE (PF-SRC)	all COBOL	COBOL sources for Client and Server. **Member** — **Description** CALC — COBOL server calculation engine. CCALCMENU — COBOL client display file. CCALCMAIN — COBOL client dialog program. CCALC — COBOL client stub derived from CobolClient1.tpl. RPCSRVI — COBOL client Broker Service derived from CobolClient2.tpl. ERXCOMM — RPC Communication Area copybook to be copied by CALCMAIN, CCALC and RPCSRVI.

Object	Type	Used by example	Description
	*FILE (PF-SRC)	all	CL procedures for compiling, binding and building. **Member** — **Description** BIND_CALC — Procedure to compile and bind the COBOL server. BIND_CCALC — Procedure to compile and bind all COBOL client modules. BIND_RCALC — Procedure to compile and bind the RPG server. CRT_C_SRV — Procedure to create modules and service programs of DEXAMPLE and EXAMPLE for the C server. CRT_CBLMOD — Compile COBOL modules.
QCSRC	*FILE (PF-SRC)	all C	Sources for C Server. **Member** — **Description** DEXAMPLE — C server generated stub. EXAMPLE — C server implementation file.
H	*FILE (PF-SRC)	all C	C header files. It contains the generated stub header CEXAMPLE.h.
QRPGLESRC	*FILE (PF-SRC)	all RPG	RPG sources for server. **Member** — **Description** CALC_RPG — RPG server calculation engine.

Step 2: Verify the RPC COBOL Client

In your EntireX network environment, the RPC server program CALC must be available for calculating figures. Examples of a CALC server program are provided in C and in COBOL. You can also access the sample CALC programs installed on your IBM i computer as described in *Step 3* and *Step 4* below.

To verify the COBOL client

1. Add the sample library EXAMPLE to your library list.
2. Call program CALCCLIENT.

   A menu similar to the following will be displayed:

```
Calculator Menu
------------------------------------------------------

Operation: ± (type + - * / to calculate or
                        type .       to terminate)
Operand 1:    _____

Operand 2:    _____

Result:  ___________


Broker-ID: localhost:1971    Server: SRV1
```

3. Specify the ID of the remote Broker and the name of the server that provides the CALC program. Specify the figures you want to compute and press **Enter**. If the Broker connection fails, you will receive an appropriate error message.

The following modules/files are bound to program CALCCLIENT:

Object	Description
CCALCMAIN	The main program logic.
CCALCMENU	The menu display file.
CCALC	The client stub derived from *CobolClient1.tpl*.
RPCSRVI	The client runtime derived from *CobolClient2.tpl*.
ERXCOMM	RPC communication area copybook.
EXA	The Broker ACI (stub).

All sources are located in the file EXAMPLE/QCBLLESRC.

Use the procedure BIND_CCALC to recompile and rebind the modules.

Note:
Program CALCCLIENT expects the server program CALC to be located in a library named EXAMPLE (as specified in the client stub CCALC). If your CALC program is located in a different library, you must adjust member EXAMPLE/QCBLLESRC(CCALC). Modify all occurrences of the string "EXAMPLE" to your library name and adapt the associated string length. Then compile and rebind the CALCCLIENT program.

Step 3: Verify the RPC Server using COBOL

To verify the RPC server under IBM i, you can use the COBOL server program `CALC` located in library EXAMPLE. When requested by a client process, it provides the four basic arithmetic operations addition, subtraction, multiplication and devision.

To verify the Server sample written in COBOL

1. Edit the RPC server configuration file `EXAMPLE/QCLSRC(RPCSRV_CFG)`.

   You must at least modify the `BrokerID=`*`Localhost`*`:1971` of the remote Broker where you want to register your server and the `ServerName=`*`SRV1`* that identifies your service.

2. Start the procedure `EXAMPLE/STR_RPCSRV`. It will submit the RPC server named `XSERVER` to a batch queue.

   If you want to extend the `SBMJOB` parameters, you can modify/recompile the procedure `EXAMPLE/QCLSRC(STR_RPCSRV)`.

   Note:
   The RPC server can only be started as a batch job for multithreading reasons. `ALWMLTTHD=*YES` is a very important parameter for allowing multiple threads. The configuration file `RPCSRV_CFG` described in the previous step will be passed to the `XSERVER`.

3. Using the IBM i command `WRKACTJOB`, you should find a job named `RPCSERVER` in your active-job list.

4. Use a calculator client process to send a request to your server. As remote client you can run a Java test generated from the Workbench *example.idl*. Or run the IBM i sample `CALCCLIENT` from your IBM i machine as described in *Step 2: Verify the RPC COBOL Client*.

   The RPC server will find and access the COBOL based sample program CALC in your library EXAMPLE and pass the computed result back to the client. Server stubs are not required for application servers written in COBOL and RPG.

   For more details on the server access logic, see *Administering the EntireX RPC Server*.

Step 4: Verify the RPC Server using C

To verify the server sample written in C

1. Edit the RPC server configuration file `EXAMPLE/QCLSRC(RPCSRV_CFG)`.

   You must at least modify the `BrokerID=`*`Localhost`*`:1971` of the remote Broker where you want to register your server and the `ServerName=`*`SRV1`* that identifies your service.

2. Start the procedure `EXAMPLE/STR_RPCSRV`. It will submit the RPC server named `XSERVER` to a batch queue.

   If you want to extend the SBMJOB parameters, you can modify/recompile the procedure `EXAMPLE/QCLSRC(STR_RPCSRV)`.

   Note:
   The RPC server can only be started as a batch job for multithreading reasons. `ALWMLTTHD=*YES` is a very important parameter for allowing multiple threads. The configuration file RPCSRV_CFG described in the previous step will be passed to the `XSERVER`.

3. Using the IBM i command `WRKACTJOB`, you should find a job named "`RPCSERVER`" in your active-job list.

4. Rename the C stub `X_DEXAMPLE` to `DEXAMPLE` and the C application `X_EXAMPLE` to `EXAMPLE`.

   Note:
   under IBM i, a service program of type `*SRVPGM` is equivalent to the UNIX term "shared library".

   For more details on the naming convention of servers and stubs written in C, see *Using the C Wrapper*.

5. Use a calculator client process to send a request to your server. As remote client you can run a Java test program generated from the Workbench example.idl.

   The RPC server will search for the shared libraries D*library* (the server stub) and *library* which contains the program functions, e.g. CALC. In our sample case, *library* denotes EXAMPLE.

   The section *Administration of the EntireX RPC Server under IBM i* describes in detail how the RPC server distinguishes between a shared library and a stubless COBOL or RPG program.