

Installing EntireX Broker Stubs under IBM i

This chapter describes how to install the Broker stub under IBM i. It covers the following topics:

- Features Currently not Supported
- Installation Steps
- Verifying the Installation of the Broker Stubs
- Upgrading from Version 5.3, 6.1 and 6.2

See *Platform Coverage* for full platform information.

The prerequisites for installing the stub are described centrally. See *IBM i Prerequisites*.

The implementation on 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* below.

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.

The installation comprises the following steps:

- Step 1: Create a Product Library
- Step 2: Copy the Installation Data Set to Disk
- Step 3: Verify the Contents of the *SAVF File
- Step 4: Restore the *SAVF File

Step 1: Create a Product Library

We recommend that you keep the EntireX Broker environment in an IBM i library named *EXX*.

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

Step 2: Copy the Installation Data Set to Disk

The product is delivered in a data set with the name `../OS400/EXAvrsp` and is stored on your EntireX installation DVD. The data set must be transferred to a *SAVF file on your machine. To do this, use the command `CRTSAVF` to create the empty save file `EXAvrsp` in a library of your choice. Then use FTP to transfer the unzipped PC data set to the save file using the FTP option "binary".

Step 3: Verify the Contents of the *SAVF File

To verify the content of the *SAVF file EXA*vrsp*, use the command DSPSAVF. This command will display the following objects:

Object	Type	Description
EXA	*SRVPGM	EntireX Broker ACI (stub).
EXXZCOMP	*SRVPGM	EntireX Compression Version 7.1.1.
TRANPOT	*SRVPGM	EntireX TCP/IP Transport Version 7.1.1.
X_SECUEXIT	*SRVPGM	EntireX Security Version 7.1.1 (to use, rename it). See <i>Using the Security Exit</i> .
ZLIB	*SRVPGM	zlib compression.
BCOC, BCOCBCL, BCOCRPG	*PGM	Client example programs in C, COBOL and RPG.
BCOS, BCOSCBCL, BCOSRPG	*PGM	Server example programs in C, COBOL and RPG.
EXABCOC	*PGM	Start BCOC client sample program (edit first; read the note under <i>Test Procedures</i>).
EXABCOCSEC	*PGM	Start BCOC client sample program with Security (edit first; read the note under <i>Test Procedures</i>).
EXABCOS	*PGM	Start BCOS server sample program (edit first; read the note under <i>Test Procedures</i>).
EXABCOSSEC	*PGM	Start BCOS server sample program with Security (edit first; read the note under <i>Test Procedures</i>).
CRT_CBLMOD	*PGM	Compile a COBOL source to an ILE module.
CRT_CMOD	*PGM	Compile a C source to an ILE module.
CRT_RPGMOD	*PGM	Compile an RPG source to an ILE module.
EXABNDPGM	*PGM	Sample: Bind user main program with the EXA service program (the Broker ACI).
EXASETENV	*PGM	Sample: Set environment variables for EXA.
EXACRTLOG	*PGM	Create a physical LOG file.
BCOC, BCOCBCL, BCOCRPG	*MODULE	Client example modules in C, COBOL and RPG.
BCOS, BCOSCBCL, BCOSRPG	*MODULE	Server example modules in C, COBOL and RPG.
EXASRC	*FILE (PF-SRC)	Example source file (sample programs and procedures).
H_EXA	*FILE (PF-SRC)	Include files for sample C programs. Rename to H before use.

Object	Type	Description
QCBLLESRC	*FILE (PF-SRC)	COBOL copybooks for sample COBOL programs.
QRPGLESRC	*FILE (PF-SRC)	RPG copybooks for sample RPG programs.
QSRVSRRC	*FILE (PF-SRC)	Export definitions for data and procedures.

Step 4: Restore the *SAVF File

To restore the *SAVF file use the command RSTLIB:

```
RSTLIB SAVLIB(EXAvrsp) DEV(*SAVF) SAVF(EXAvrsp) RSTLIB(EXX)
```

After you have restored the save file, the product library EXX should at least contain the objects listed in step 3.

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

The EntireX Broker ACI for IBM i is now installed and ready to be bound to a user-developed application environment.

Verifying the Installation of the Broker Stubs

This package contains test programs (BCOS, BCOC) and example programs for binding (LINKPGM).

Test Programs

There are two types of programs to test the connection to the remote Broker.

- BCOS, BCOSCBL and BCOSRPG

These programs act as servers. When they have been registered with the remote Broker, they wait for incoming calls from the client programs BCOC, BCOCBL and BCOCRPG, respectively, and respond immediately.

- BCOC, BCOCBL and BCOCRPG

These programs act as clients. They attach in non-conversational mode to one service and send data to the registered server programs BCOS, BCOSCBL and BCOSRPG, respectively.

The file EXASRC contains the sources of the programs listed above; they are available in C, COBOL (...CBL) and RPG (...RPG).

Prerequisite for the test is a running remote Broker. To proceed, you need to know the Broker ID (brokerid), the port number (portno) and, optionally, the API (ACI) version of your Broker kernel (APIversion).

1. In two interactive sessions, set the *CURLIB to library EXX.
2. Check that the log file in library EXX exists. If it does not exist, create it using the example program EXACRTLOG.
3. In session 1, execute the command:

```
Call BCOS ('-b brokerid: portno ' '- APIversion ')
```

4. In session 2, execute the command:

```
Call BCOC ('-b brokerid: portno ' '- APIversion ')
```

The steps above use the C samples, the samples for COBOL and RPG (see the program names above) can be used with the same parameters. The COBOL and RPG samples have less screen output than the C sample.

With Broker Kernel API Version 6 and the Broker ID Localhost, the screen output of BCOC will look similar to this:

```
CLIENT: ETB Version 2
CLIENT: Number of repeats (10)
CLIENT: Broker ID Localhost:3930
CLIENT: Broker ID pcbal2:6099
CLIENT: Using default class ACLASS
CLIENT: Using default server ASERVER
CLIENT: Using default service ASERVICE
CLIENT: Maximum stub ACI version: 7
CLIENT: Maximum kernel ACI version: 6
CLIENT: Kernel Version 6.2.1.00 Platform UNIX
```

```
CLIENT: Using ACI version: 6
Request issued (9, 2037)
Request issued (8, 2037)
Request issued (7, 2037)
Request issued (6, 2037)
Request issued (5, 2037)
Request issued (4, 2037)
Request issued (3, 2037)
Request issued (2, 2037)
Request issued (1, 2037)
Request issued (0, 2037)
CLIENT: Test successfully performed
Press ENTER to end terminal session.
```

The screen output of BCOS will look similar to this:

```
SERVER: ACI Version 6
SERVER: Number of repeats (10)
SERVER: Broker ID Localhost:3930
SERVER: Using default class ACLASS
SERVER: Using default server ASERVER
SERVER: Using default service ASERVICE
SERVER: Maximum stub ACI version: 7
SERVER: Maximum kernel ACI version: 6
SERVER: Kernel Version 6.2.1.00 Platform UNIX
```

```
SERVER: Using ACI version: 6
Incoming data (2037, 9)
Incoming data (2037, 8)
```

```
Incoming data (2037, 7)
Incoming data (2037, 6)
Incoming data (2037, 5)
Incoming data (2037, 4)
Incoming data (2037, 3)
Incoming data (2037, 2)
Incoming data (2037, 1)
Incoming data (2037, 0)
SERVER: Test successfully performed
Press ENTER to end terminal session.
```

If it does, the test with the remote Broker was successful.

Test Procedures

To perform the test programs, you can also modify and use the sample procedures located in the source file EXASRC:

Procedure	Description	Programs called.
EXABCOC	Client test procedure	<u>BCOC</u> , BCOCCL, BCOCRPG.
EXABCOCSEC	Client test procedure with Security parameters	<u>BCOC</u> only.
EXABCOS	Server test procedure	<u>BCOS</u> , BCOSCL, BCOSRPG.
EXABCOSSEC	Server test procedure with Security parameters	<u>BCOS</u> only.

Note:

By default, all procedures call the "C" flavor of the client and server programs, i.e. BCOC and BCOS.

To call the relevant COBOL (BCOCCBL, BCOSCBL) or RPG (BCOCRPG, BCOSRPG) programs, you must modify the procedures listed above accordingly. After having adjusted the Broker ID, Broker Version and Security parameters, you must compile the sources and bind the created modules to executable *PGM programs. For compiling, use the procedure CRT_CMOD, CRT_CBLMOD or CRT_RPGMOD. For binding, use the procedure EXABNDPGM.

All sample programs incorporate the ACI Broker control block definitions during compilation. The relevant include file and copybooks are located in:

Include source file	Copy member	Description
H	ETBCDEF	C ACI definitions for BCOS and BCOC.
	ETBCINF	Broker API definitions for Command/Info Services using C. See Broker Command and Information Services.
QCBLLESRC	COBDEF	COBOL ACI definitions for BCOSCBL and BCOCBL.
	COBINF	Broker API definitions for Command/Info Services using COBOL. See Broker Command and Information Services.
QRPGLESRC	RPGDEF	RPG ACI definitions for BCOSRPG and BCOCRPG.

Upgrading from Version 5.3, 6.1 and 6.2

Step 1: Make a Backup

Make a backup of the existing version in library EXX.

Step 2: Install Latest EntireX ACI Version

Perform the steps listed in *Installation Steps* above.

All program and sample files in the library EXX will be replaced.

Step 3: Bind the Application with latest ACI Version

All application server programs that use the Broker stub (ACI) must be rebound with the server program EXA.

To perform this step, you can use the command CRTPGM (see CL example EXABNDPGM).

Using command DSPPGMREF (Display Program Reference) you can display the service programs that are currently referenced to your application.

If you use Natural RPC, you must rebound the Natural Broker Wrapper program NXWRAPPER located in your library NXWvrs.