

# Administration of Broker Stubs under IBM i

This chapter covers the following topics:

- Using the Security Exit
  - Tracing for Broker Stubs under IBM i
- 

## Using the Security Exit

When you start a Broker ACI application program, the ACI service program searches your library list for the Security Exit service `SECUEEXIT`. If it is found, it will be used automatically.

### Using the Security Exit supplied by Software AG

In this delivery package, a Security Exit named `X_SECUEEXIT` is provided by Software AG. However, this file does not match the required name `SECUEEXIT`. So by default, the Broker ACI runs without using a Security Exit.

#### To use the Software AG-supplied Security Exit Program

1. Rename the \*SRVPGM-type program `X_SECUEEXIT` to `SECUEEXIT` and
2. Start your user application. The bound Broker ACI will find and use the exit.

For detailed information, see Sample Security Exits for Broker Security.

### Creating your own Security Exit Program

A user-supplied security exit must meet the following minimum requirements:

Entry points	etbueva etbupre
Service program exports	STRPGMEXP PGMLVL(*CURRENT) LVLCHK(*NO) EXPORT SYMBOL('etbupre') EXPORT SYMBOL('etbueva') ENDPGMEXP  Sample: QSRVSRCLIB(SECUEEXIT) Type *BND
Calling convention	... int etbueva(ETBCB *pEtbCb, void *pEncBuf, void *pReserved, char *pErrTxt) ... int etbupre(ETBCB *pEtbCb, void *pSndBuf, void *pReserved, char *pErrTxt)  Sample source: EXASRC(SECUEEXIT)
Create a Service Program for a security user exit	Sample source: EXASRC(EXACRTSXIT) This CL procedure creates module SECUEEXIT from a C source and binds it to the service program SUEXIT.

### ➤ To create a security exit program

1. Write the source code, include the necessary encryption and decryption algorithms. The supplied C example SECUEEXIT provides a framework for the basic functionality.
2. Create a source file of the type \*BND for the service program exports (see *Creating your own Security Exit Program*).
3. Compile the source code.
4. Create the service program.

See the CL Program sample EXACRTSXIT that contains the relevant commands (CRTCMOD and CRTSRVPGM) for compiling and creating a service program. In that example a service program named SUEXIT is created.

5. Usually, the Software AG-supplied security exit is named X\_SECUEEXIT. If you have previously renamed it SECUEEXIT, rename it back now to X\_SECUEEXIT for backup reasons.

Then rename the security exit service program SUEXIT you have just created to SECUEEXIT.

6. Test the application. When you start your application program, the Broker ACI automatically finds and uses the Security Exit SECUEEXIT.

The samples above are based on C examples. For other languages such as ILE COBOL and ILE RPG, use the respective compiler.

## Tracing for Broker Stubs under IBM i

To request a log file from the Broker stub, the environment variable `ETB_STUBLOG` must be set. The value of this variable defines how detailed the log will be.

The following table describes the trace values for `ETB_STUBLOG`:

Trace Value	Trace Level	Description
0	NONE	No tracing.
1	STANDARD	Traces initialization, errors, and all ACI request/reply strings.
2	ADVANCED	Used primarily by system engineers, traces everything from level 1 and provides additional information - for example the Broker ACI control block - as well as transport information.
3	SUPPORT	This is full tracing through the stub, including detailed traces of control blocks, message information, etc.

### > To evaluate error conditions

1. Set the environment variable: run the program `EXASETENV` or use the command:

```
ADDENVVAR ENVVAR(ETB_STUBLOG) VALUE(3)
```

To change the value of the variable, use the command `WRKENVVAR` or change and recompile the source file in `EXASRC`.

2. Re-run the example programs.

The member names in the file `LOG` are created with the prefix `ETB` and the six-digit process ID (`ETBpppppp`).

**Note:**

The file will be overwritten if you restart your application in the same session.