Administering EntireX RPC-ACI Bridge

The RPC-ACI Bridge enables RPC-based client applications to be used with ACI servers.

This chapter covers the following topics:

- Customizing the RPC-ACI Bridge
- Configuring the RPC Server Side
- Configuring the ACI Client Side
- Starting the RPC-ACI Bridge
- Stopping the RPC-ACI Bridge
- Application Identification

Customizing the RPC-ACI Bridge

For the setup of the RPC-ACI Bridge there are

- a configuration file and
- scripts to start the RPC-ACI Bridge.

Location of the RPC-ACI Bridge

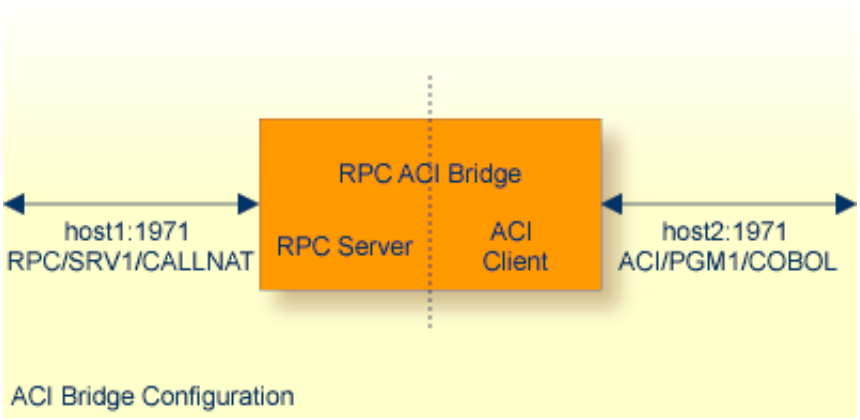
The RPC-ACI Bridge is contained in the file *entirex.jar*.

The Configuration File

The default name of the configuration file is *entirex.rpcacibridge.properties*. The RPC-ACI Bridge searches for this file in the current working directory.

You can set the name of the configuration file with `-Dentirex.server.properties=<your file name>` with "/" as file separator.

The configuration file contains the configuration for both parts of the RPC-ACI Bridge.



To set up the RPC-ACI Bridge

1. Use the RPC server agent of the System Management Hub.
2. Add the RPC-ACI Bridge as an RPC server.

   See *Administering the EntireX RPC Servers using System Management Hub* under UNIX | Windows for details.

   Or:
   Use the scripts to start the RPC-ACI Bridge.

   Under Windows use *jrpcacibridge.bat* in the folder *bin* to start the RPC-ACI Bridge. You may customize this file.

   Under UNIX use *jrpcacibridge.bsh* in the folder *bin* to start the RPC-ACI Bridge. You may customize this file.

   Both scripts use the configuration file *entirex.rpcacibridge.properties* in the folder *etc*.

Configuring more than one RPC-ACI Bridge

If you configure more than one RPC-ACI Bridge that connect to the same EntireX Broker, the following items must be distinct:

- The user for the ACI client side (property `entirex.rpcacibridge.userid`).
- The trace output file (property `entirex.server.logfile`).
- The monitor port for SMH (property `entirex.server.monitorport`).
- The log for the Windows Service (property `entirex.server.serverlog`).
- The trace output file of the SMH agent for RPC servers.

Configuring the RPC Server Side

The RPC server side of the RPC-ACI Bridge is configured like the Java RPC Server. The RPC-ACI Bridge uses the properties that start with "entirex.server".

The RPC server side can adjust the number of worker threads to the number of parallel requests. Use the properties `entirex.server.fixedservers`, `entirex.server.maxservers`, `entirex.server.minservers` to configure this scalability. If `entirex.server.fixedservers=yes`, the number of `entirex.server.minservers` is started and the server can process this number of parallel requests. If `entirex.server.fixedservers=no`, the number of worker threads balances between `entirex.server.minservers` and `entirex.server.maxservers`. This is done by a so-called attach server thread. On startup, the number of worker threads is `entirex.server.minservers`. If more than `entirex.server.minservers` are waiting for requests, a worker thread stops if its receive call times out. The timeout period is configured with `entirex.server.waitserver`.

Alternatively to the properties, you can use the command-line option. The command-line options have a higher priority than the properties set as Java system properties and these have higher priority than the properties in the configuration file.

Name	Command-line Option	Default Value	Explanation
`entirex.server.brokerid`	`-broker`	`localhost : 1971`	Broker ID
`entirex.server.codepage`	`-codepage`		The codepage the server uses. Permitted values are the name of the codepages the JVM supports. Use the value LOCAL when the default codepage of the JVM should be used. See *Internationalization with EntireX* for details.
`entirex.server.compresslevel`	`-compresslevel`	0 (no compression)	`BEST_COMPRESSION` 9 `BEST_SPEED` 1 `DEFAULT_COMPRESSION` -1, mapped to 6 `DEFLATED` 8 `NO_COMPRESSION` 0 `N` 0 `Y` 8
`entirex.server.encryptionlevel`	`-encryption`	`0`	Encryption level (if Broker is version 6.1.1 or higher. Valid values: 0,1,2).
`entirex.server.environment`			Can be used in a user-written translation exit of the broker. See `BrokerService`, `setEnvironment(java.lang.String)` (EntireX Java ACI).
`entirex.server.fixedservers`		`no`	If "no", use an attach server thread to manage worker threads, otherwise run minimum number of server threads. See properties `entirex.server.maxservers`, `entirex.server.minservers`.
`entirex.server.logfile`	`-logfile`		Path and name of the trace output file.
`entirex.server.maxservers`		`32`	Maximum number of worker threads.
`entirex.server.minservers`		`1`	Minimum number of server threads.

Name	Command-line Option	Default Value	Explanation
entirex.server.monitorport	-smhport	0	The port where the server listens for commands from the System Management Hub (SMH). If this port is 0, no port is used and the management by the SMH is disabled.
entirex.server.name			The name of the server.
entirex.server.password	-password		The password for secured access to the Broker. The password is encrypted and written to the property entirex.server.password.e. To change the password, set the new password in the properties file (default is *entirex.rpcacibridge.properties*). To disable password encryption set entirex.server.passwordencrypt=no. Default for this property is "yes".
entirex.server.properties	-propertyfile	entirex.rpcacibridge.properties	The file name of the property file.
entirex.server.restartcycles	-restartcycles	15	Number of restart attempts if the Broker is not available. This can be used to keep the Java RPC Server running while the Broker is down for a short time.
entirex.server.security	-security	no	no/yes/auto/Name of BrokerSecurity object.
entirex.server.serveraddress	-server	RPC/SRV1/CALLNAT	Server address
entirex.server.serverlog	-serverlog		Name of the file where start and stop of worker threads is logged. Used by the Windows RPC Service.
entirex.server.userid	-user	JavaServer	The user ID of the Broker for RPC. See entirex.server.password.
entirex.server.verbose	-verbose	no	Verbose output to standard output yes/no.
entirex.server.waitattach		600S	Wait timeout for the attach server thread.
entirex.server.waitserver		300S	Wait timeout for the worker threads.
entirex.timeout		20	TCP/IP transport timeout. See *Setting the Transport Timeout* under *Writing Advanced Applications - EntireX Java ACI*.
entirex.trace	-trace	0	Trace level (1,2,3).

Configuring the ACI Client Side

These properties are used to configure the connection to the Broker for ACI.

Alternatively, you can use the command-line option. The command-line options have a higher priority than the properties set as Java system properties and these have higher priority than the properties in the configuration file

Name	Command-line Option	Default Value	Explanation
`entirex.rpcacibridge.brokerid`	`-acibroker`	`localhost`	Broker ID of the Broker for ACI.

`entirex.rpcacibridge.compresslevel`	`-acicompresslevel`	`0` (no compression)	Permitted values (you can enter the text or the numeric value):
			`BEST_COMPRESSION` `9`
			`BEST_SPEED` `1`
			`DEFAULT_COMPRESSION` `-1`, mapped to `6`
			`DEFLATED` `8`
			`NO_COMPRESSION` `0`
			`Y` `8`

`entirex.rpcacibridge.encryptionlevel`	`-aciencryption`	`0`	Encryption level (if Broker is version 6.1.1 or higher. Valid values: 0,1,2).
`entirex.rpcacibridge.marshalling`	`-acimarshalling`		This is for arrays of groups. Set this property to "cobol" if the ACI server is a COBOL program. Set this property to "natural" if the ACI server is a Natural program. Default is " ", which lets the RPC client determine the marshalling.
`entirex.rpcacibridge.password`	`-acipassword`		The password of the Broker for ACI. The password is encrypted and written to the property entirex.server.password.e. To change the password, set the new password in the properties file (default is *entirex.rpcacibridge.properties*). To disable password encryption set `entirex.server.passwordencrypt=no`. Default for this property is "yes".
`entirex.rpcacibridge.security`	`-acisecurity`	`no`	no/yes/auto/Name of BrokerSecurity object.
`entirex.rpcacibridge.serveraddress`	`-aciserver`	`ACLASS/ASERVER/ASERVICE`	Server Address for the Broker for ACI.
`entirex.rpcacibridge.trace`	`-acitrace`	`No`	If set to "yes", additional trace output (exception stack-traces and request and reply buffers) is generated.
`entirex.rpcacibridge.userid`	`-aciuser`	Value of system property user.name .	The user ID of the Broker for ACI. Use different user IDs for different RPC-ACI Bridges on the same Broker.
`entirex.rpcacibridge.waittime`		`0S`	The wait time for receive requests. Permitted values are *nS|nM|nH*, where *n* is the number of seconds or minutes or hours.

Starting the RPC-ACI Bridge

To start the RPC-ACI Bridge

- Use the script *jrpcacibridge* in the folder *bin* to start the RPC-ACI Bridge. You may customize this file.

  Or:
  Use the RPC server agent in the System Management Hub to configure and start the RPC-ACI Bridge.

  See *Administering the EntireX RPC Servers using System Management Hub* under UNIX | Windows for details.

On Windows you can start the RPC-ACI Bridge as a Windows Service. The installation of the service is similar to the installation of the Java RPC Server. See *Running the Java RPC Server as a Windows Service*.

Stopping the RPC-ACI Bridge

To stop the RPC-ACI Bridge

- Use the RPC server agent in the SMH to stop the RPC-ACI Bridge.

  Or:
  Use the agent for the Broker. Use `Deregister` on the service, specified with the property `entirex.server.serveraddress`.

Application Identification

The application identification is sent from the RPC-ACI Bridge to the Broker. It is visible with Broker Command and Info Services.

The identification consists of four parts: name, node, type, and version. These four parts are sent with each Broker call and are visible in the trace information.

For the RPC-ACI Bridge these values are:

Application name:	`ANAME=RPC-ACI Bridge`
Node name:	`ANODE=<host name>`
Application type:	`ATYPE=Java`
Version:	`AVERS=9.0.0.0`